DATA FUSION AND DISTRIBUTED ROBOTIC PERCEPTION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jonathan R. Schoenberg

January 2012

DATA FUSION AND DISTRIBUTED ROBOTIC PERCEPTION

Jonathan R. Schoenberg, Ph.D.

Cornell University 2012

This thesis explores data fusion and distributed robotic perception through a series of theoretical developments, analyses and experiments. First, a GSF with component extended Kalman filters (EKF) is proposed as an approach to localize an autonomous vehicle in an urban environment with limited GPS availability. The GSF is used because of its ability to represent the posterior distribution of the vehicle pose with better efficiency (fewer terms, less computational complexity) than a corresponding bootstrap particle filter with various numbers of particles due to the interaction with measurement hypothesis tests. A series of in-depth empirical studies are performed using 37 minutes of recorded data from Cornell University's autonomous vehicle driven in an urban environment, including a 32 minute GPS blackout.

Second, a distributed grid-based terrain mapping algorithm using Gaussian Mixture Models is developed for use in tree connected and arbitrary connected sensor networks. The distributed data fusion rules are developed that operates directly on the sufficient statistics summarizing the grid-cell height and uncertainty. The distributed grid-based terrain mapping algorithms is demonstrated in an experimental environment involving 8 autonomous robots operating in an indoor environment for 120 seconds.

Third, an algorithm to segment 3D points in dense range maps generated from the fusion of a single optical camera and a multiple emitter/detector laser range finder is presented. The algorithm is demonstrated on data collected with the

Cornell University DARPA Urban Challenge vehicle.

Finally, two information theoretic procedures for fusing multiple distributions with unknown correlation are developed. The first approach developed is Entropy Weighted Chernoff fusion; this fusion procedure biases the WEP fusion weight towards the distribution with the lowest entropy. An information loss for the WEP conservative fusion rule is introduced and an approximation derived by computing the Kullback-Leibler divergence between the Naive Bayes and WEP fused distributions. The approximation is minimized for the second fusion approach: Minimum-Information-Loss fusion; the procedure generates the least conservative fused distribution in the family of WEP results. Experimental results include the fusion of multiple occupancy grid maps over an optimally connected sensor network, demonstrating consistent map estimates.

BIOGRAPHICAL SKETCH

Jonathan R. Schoenberg earned his Bachelor of Science degree in Electrical and Computer Engineering from Cornell University in January 2003. He completed his Masters of Engineering degree in Electrical and Computer Engineering from Cornell University in May 2003. Jonathan earned a Masters of Science in Aerospace Engineering from Cornell University in August 2009. He is completing a Ph.D. in Aerospace Engineering at Cornell University under Professor Mark Campbell.

To Katie.

ACKNOWLEDGEMENTS

Thank you to Professor Mark Campbell for the many hours of fruitful and engaging discussions that made the process stimulating and enjoyable.

Thank you to my committee: Professors Mark Psiaki, Hadas Kress-Gazit and Phaedon-Stelios Koutsourelakis for valuable inputs and guidance in class work and in research.

Thank you to the members of the Autonomous Systems Lab and DARPA Urban Challenge Team who made possible the data collections performed in this thesis.

Thank you to Northrop Grumman Corporation for the opportunity to take on this challenge through the Electronic Systems Scholars Program.

Thank you to ARO Grant #W911NF-09-1-0466 and NSF Cyber-Physical Systems Grant #CNS-0931686 for additional funding when the process took a little longer.

Thank you to The Library for always being open at just the right times.

Thank you to the friends I have made along the way and the quality distractions you provided at all the right times.

Thank you to my extended family for always supporting us on the path we chose to wander.

Thank you to Dutch, Anabel and Bailey for never questioning the decisions we make.

Thank you to Katie for all the adventures, love and support; without you, this thesis would not fit in a box.

Thank you to Marie Wren for motivation and a fresh perspective.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This thesis addresses data fusion and distributed robotic perception through a series of papers. Mobile robotics are moving into society in areas including household chores, driving and agriculture. The proliferation will continue as actuators and sensors continue to fall in cost. Additional, data networks are expanding in capacity and coverage. The data networks include Wifi network and cellular data networks capable of connecting mobile robots together in an unprecedented fashion. As mobile robotics and data networks continue to evolve, the goal is to enable always-on and always-collaborating robotics.

To enable this goal, a question to address is what is the best way to fuse and share information collected from networked mobile robots. The robotics today operate in isolation and are responsible for obtaining, processing and acting on self-collected information. In a mobile robot network, the agents have the ability to share data, decisions, and processed results. This thesis begins to address the goal of always-on and always-collaborating robotics and focuses on the question of how to fuse and share the information available to each agent in the network.

The first area to address is a methodology for multi-sensor fusion and multi-modal estimation. The multi-sensor fusion question arises as individual robots contain different sensors that can be used for a common process, such as localizing a vehicle. The multi-modal aspect arises, because sensor error models are not well-modeled by a single Gaussian distribution (Anderson and Moore, 1979). Therefore, the multi-sensor fusion and multi-modal estimation problem is solved via a formal, real-time Gaussian Sum Filter (GSF) using a novel combination of a GSF with Runnalls' condensation algorithm. The algorithm properties are inves-

tigated for representational efficiency of the posterior distribution compared to a particle filter. The work is in Chapter 2 and is derived from (Schoenberg et al., 2009) and (Schoenberg et al., 2012). The compact posterior representation enables networked robots to efficiently share information about their own localization for probabilistic fusion in a sensor network.

The second area to address is a technique for sharing and fusing rich probabilistic environment representations. Mobile robots can independently collect dense probabilistic representations of the environment that are useful for planning and exploration (Miller and Campbell, 2006). The goal is to share the environmental maps with other agents to expand the domain information contained at each agent. To efficiently share the information, a minimal set of data should be exchanged that captures the content at each agent. Additionally, a robust and flexible network configuration such as an optimally connected network is desired to avoid brittle topologies such as a centralized configuration that have a single point of failure. Working towards these goals, a distributed grid-based terrain mapping algorithm for multi-robot mapping is developed. The work is in Chapter 3 and is derived from (Schoenberg and Campbell, 2009) and (Schoenberg and Campbell, 2012b).

The third area to address is techniques to exploit sensor modality for higher-level tasks such as object tracking and path planning. Different sensors capture different types of information about the environment where a mobile robot is operating. In particular, vision sensors capture rich optical intensity information projected into a 2D image. By contrast laser sensors capture sparse 3D information and intensity in the IR domain. The efficient and effective combination of the two sensor modalities is explored for segmentation. An effective segmentation

of an scene where a mobile robot is operating clusters like objects or surfaces together, but separates distinct objects. The segments are efficient building blocks for target tracking or path planning algorithms. An algorithm to efficiently and accurately segment dense range maps produced by the fusion of camera and range data in a complete urban environment in near real-time is developed. The work is in Chapter 4 are is derived from (Schoenberg et al., 2010).

The final area to address is consistent data fusion with unknown correlation in robust sensor networks. As mobile robots become ubiquitous and share information freely on an unconstrained network, the problem of rumor propagation persists. Rumor propagation is the transmission and acceptance of unverified or falsely attributed information. In the case of mobile robots, rumor propagation occurs when the same data completes a cycle around the sensor network, but is accepted as novel information. If the repeated data is treated as novel information and fused without regard for pedigree, the estimate using that information becomes optimistic and inconsistent. An alternative approach is to data-tag all information and only count novel unseen data, but this leads to transmission redundancies and inefficient communication utilization. It is possible to fuse data with unknown correlation in a conservative, but consistent manner. These data fusion schemes discount exclusive information in each estimate, but maintain all common information. The approaches have drawbacks that include the inability to adjust the fusion based on the information content of each distribution and a guaranteed conservative fusion if the two distributions are uncorrelated. Therefore, two novel generalized fusion approaches for fusing distributions with unknown correlation are developed. The two approaches are Entropy Weighted Chernoff Fusion and Minimum-Information-Loss Fusion. The work is in Chapter 5 and is derived from (Schoenberg and Campbell, 2012a).

1.1 Thesis Contributions

The first section focuses on data fusion of multiple sensors to localize an autonomous vehicle in a sparse GPS environment and addresses a methodology for multi-sensor fusion and multi-modal estimation. The contributions from Chapter 2 are:

- Formal, real-time GSF implementation using novel combination of a GSF with Runnalls' condensation algorithm to solve fundamentally multi-model problem of localizing an autonomous vehicle in a sparse GPS environment
- Adaptive number of Gaussians in the GSF posterior using the MMSE estimate
- Robust empirical evaluation of performance and computation of the RT-GSF
- Statistical demonstration GSF better represents the posterior distribution of the vehicle pose compared to the bootstrap particle filter
- Analysis of the number of terms in the particle filter posterior distribution shows small term Gaussian mixture adequately describes the posterior distribution in real-time
- RT-GSF outperforms the particle filter regardless of location on the map due to interaction of the measurement hypothesis tests

The second section focused on distributed robotic perception in the area of the collaborative terrain mapping and develops a technique for sharing and fusing rich probabilistic environment representations. The contributions from Chapter 3 are:

- Developed two distribution grid-based terrain mapping algorithms for use in multi-robot mapping

- Develop technique to utilize the channel filter based on compact information sets representing in-cell height estimate
- Develop novel approach for consistent, but conservative fusion for arbitrary network topologies based on fusion of sufficient statistics
- Evaluate information gain using channel filter based on tree topology
- Evaluate distributed mapping technique on optimally connected network to generate quality terrain maps with consistent, but conservative fusion
- Develop two novel metrics for information content in a network performing distributed terrain mapping
- Derive bound on information loss when performing distributed data fusion
- Evaluate map dependent location of information loss for use in planning exploration vs verification activities

The third section focused on data fusion and robotic perception in the area of fusion of range and camera data and the segmentation of the scenes for higher-level processing and the exploiting sensor modality. The contributions from Chapter 4 are:

- Efficiently and accurately segment dense range maps produced by fusion of camera and range data in a complex urban environment in near real-time
- Novel combination of Markov Random Field to create dense range maps and segmentation using range, color intensity and surface normals
- Demonstrate efficacy of using dense range maps as opposed to incorporating color intensity and surface normal estimates into the sparse data

The fourth section focused generalized distributed data fusion approaches for arbitrary distributions with application to fusion of multi-robot occupancy grid over arbitrary networks and addresses consistent data fusion with unknown correlation. The contributions from Chapter 5 are:

- Develop two novel generalized fusion approaches for fusing distribution with unknown correlation, Entropy Weighted Chernoff Fusion and Minimum-Information-Loss Fusion
- Develop information loss metric resulting from fusion for arbitrary distributions
- Derive an approximate information loss metric for arbitrary distributions
- Analysis of different fusion rules for the Bernoulli distribution
- Demonstration of applicability of generalized fusion rules to distributed mapping using occupancy grids

CHAPTER 2

POSTERIOR REPRESENTATION WITH A MULTI-MODAL LIKELIHOOD USING THE GAUSSIAN SUM FILTER FOR LOCALIZATION IN A KNOWN MAP

Abstract

A Gaussian Sum Filter (GSF) with component extended Kalman filters (EKF) is proposed as an approach to localize an autonomous vehicle in an urban environment with limited GPS availability. The GSF uses vehicle relative vision-based measurements of known map features coupled with inertial navigation solutions to accomplish localization in the absence of GPS. The vision-based measurements have multi-modal measurement likelihood functions that are well represented as a weighted sum of Gaussian densities. The GSF is used because of its ability to represent the posterior distribution of the vehicle pose with better efficiency (fewer terms, less computational complexity) than a corresponding bootstrap particle filter with various numbers of particles due to the interaction with measurement hypothesis tests. The Expectation-Maximization algorithm is used offline to determine the representational efficiency of the particle filter in terms of an effective number of Gaussian densities. By comparison, the GSF, which uses an iterative condensation procedure after each iteration of the filter to maintain real-time capabilities, is shown through a series of in-depth empirical studies to more accurately maintain a representation of the posterior distribution than the particle filter using 37 minutes of recorded data from Cornell University's autonomous vehicle driven in an urban environment, including a 32 minute GPS blackout.

2.1 Introduction

Autonomous vehicles provide opportunities to remove humans from operating in dangerous civilian and military scenarios. The autonomy relies critically on accurate localization of the vehicle in diverse environments, including urban environments where absolute position information is not available from a Global Navigation Satellite System (GNSS). The urban environment is challenging for reliable position estimation from GNSS signals due to multi-path reflections, obstruction of direct path signals and absolute denial. In the absence of absolute position information, autonomous vehicles rely on dead reckoning from an inertial navigation system (INS) to localize the vehicle. Unfortunately, small errors in the INS solution accumulate into large position deviations after a few minutes, which in turn, prevent the vehicle from localizing itself within a lane.

The PosteriorPose algorithm (Miller and Campbell, 2008) demonstrates a bootstrap particle filter (PF) to be effective in providing map relative localization in the absence of absolute measurements. The techniques do not constrain the vehicle to the road, but use vision data to observe features in the known map for localization. The map-aided localization problem is fundamentally multi-modal because the segmented vision data provides measurements of closely-space landmarks with unknown correspondence to the known map. This is similar to the data association problem in SLAM (Bailey and Durrant-Whyte, 2006), where incorrect assignments cannot be reversed and can lead to failure of the SLAM algorithm. The problem solved here differs from SLAM because an accurate map enables prediction of data assignment mistakes that manifest as multiple modes in the measurement and posterior densities. Additionally, the state vector is small because the map is known; multiple modes can be maintained in the posterior density that would be

less feasible in a SLAM approach with a large unknown map.

The PosteriorPose algorithm (Miller et al., 2011), (Miller and Campbell, 2008) utilizes the bootstrap PF as the approximate recursive Bayesian estimation algorithm to handle nonlinear vehicle motion and multi-modal vision-based measurement functions. The PF relies on a point mass representation of the posterior density; there are several drawbacks of the PF including finite support over the posterior state space, particle degeneracy, difficulty in selecting an appropriate importance density, and sample impoverishment after resampling (Arulampalam et al., 2002). Despite the successful real-time implementation in the DARPA Urban Challenge (DUC) (Miller and Campbell, 2008), the PosteriorPose algorithm still suffers from these drawbacks. The Gaussian Sum Filter (GSF) is proposed as a solution to the map-aided localization problem to avoid the drawbacks of the PF while handling the multiple modes of the vision-based measurement and posterior densities.

The theoretical convergence results of the particle filter have been studied extensively (Crisan and Doucet, 2002) since the bootstrap particle filter was introduced (Gordon et al., 1993). The accepted conclusion is that given enough particles, the bootstrap particle filter will converge to the true optimal filter posterior density (Crisan and Doucet, 2002). Unfortunately, this requires the assumption that the importance density (the prior in the case of the bootstrap particle filter) is 'close' to the posterior and the particle variance introduced in the resampling step used to avoid particle impoverishment is 'small.' Improved importance densities and resampling schemes (Doucet and Johansen, 2009), (Douc and Cappe, 2005) have been proposed to combat these issues. In this paper, the posterior representational efficiency of the bootstrap particle filter, which was run in real-time for the

PosteriorPose algorithm (Miller and Campbell, 2008) is evaluated and compared to a GSF solution.

The availability of recognizable street markings around an autonomous vehicle motivates the development of map-aided localization techniques. The Map-Aided GPS (MAGPS) method (Syed and Cannon, 2005) is proposed where road segment information derived from a robust map-matching technique is tightly coupled into the GPS solution as a set of constraints. Also addressing the problem of self-localization, (Cui and Ge, 2003) and (Fouque and Bonnifait, 2008) propose techniques that tightly couple GPS signals with a known map. Cui and Ge strictly restrict motion to the map, while Fouque and Bonnifait treat the road as a noisy measurement. Localization in the context of simultaneous localization and mapping (SLAM) by constraining the platform to a set of known road segments is shown (Wijesoma et al., 2005). Other areas of map-aided estimation include (Cheng and Singh, 2007) who perform target tracking with a particle filter for ground moving targets constrained to a road network and (Levinson et al., 2007) who built a map of ground reflectivity in infrared and used a particle filter to localize the vehicle in that map. In addition, (Fosbury et al., 2007) use terrain potential functions based on trafficability to constrain the motion of an off-road ground mover during track. The Stanford University entry into the DARPA Urban Challenge (Montemerlo et al., 2008) utilized lidar sensor reflectivity and curb identification to perform map relative localization with a one-dimensional histogram of the vehicle's lateral offset within a lane. Overhead map imagery was used to correct a SLAM-based locally accurate trajectory generated from relative bundle adjustment by matching local vision images to the known map (Napier et al., 2010), however false data associations were identified as a cause of divergent results. Each of these algorithms make strong assumptions about correlations between vehicle motion along

the known road map and correctness about data association. Unfortunately, autonomous vehicles are required to operate free from the strict constraints of road maps and therefore require localization techniques that account for this freedom of motion off the map. The motivating example may be a road block or disabled car that requires the autonomous vehicle to pass on the wrong side of the road, complete an impromptu U-turn, or drive off the road around the obstacle.

The Gaussian Sum Filter (GSF) has been used to solve nonlinear recursive Bayesian estimation problems since it was introduced (Sorenson and Alspach, 1971), this describes the GSF by representing the desired a priori, transition, measurement, and posterior densities as a summation of component Gaussians. The primary problem with the GSF is computational intractability: the number of component Gaussians grows geometrically with each iteration of the GSF. (Alspach and Sorenson, 1972) recognized this failure to manage the number of terms as a limit in the utility of the GSF, and proposed combining components with equal means and covariances and eliminating terms with neglectable weights.

Several tractable approaches to managing the growing number of component Gaussians in recursive GSF implementations have arisen as solutions to probabilistic data association in target tracking, where the GSF has been used to account for highly-maneuvering targets amidst considerable clutter (Kozak and Maybeck, 2006), (Maybeck and Smith, 2005), (Williams, 2003). The GSF framework has been applied to multi-modal probabilistic data association problem in target tracking (Bar-Shalom and Tse, 1975). The Gaussian Sum framework is used for joint probabilistic data association in multi-target tracking in clutter (Fortmann et al., 1983). The probabilistic data association techniques manage the number of terms by collapsing the Gaussian mixture to a single Gaussian at each iteration. Outside

of the data association problem, (Peach, 1995) and (Kronhamn, 1998) apply the GSF to bearing-only target tracking and each use a pruning approach to manage the number of terms in the posterior density. Similarly, (Kwok et al., 2005) use the GSF to solve the initialization problem in bearing-only SLAM. The mixture reduction is performed by truncation, and component Gaussians are removed based on a sequential probability ratio test. As a computational alternative and approximation to the GSF, (Sola et al., 2005) propose Federated Information Sharing (FIS) to solve the problem of the computational intractability of the GSF for undelayed initialization in bearing-only SLAM and for SLAM with panoramic vision (Lemaire and Lacroix, 2007).

Another area of the literature which is relevant is mixture reduction algorithms, which propose to reduce the number of mixands in a Gaussian sum mixture using some metric, but not in real-time. Two mixture reduction algorithms, joining and clustering, are presented (Salmond, 1988) as alternatives to collapsing to a single Gaussian when using Gaussian sums for data association in single target tracking. A different mixture reduction technique based on the integrated square distance (Williams and Maybeck, 2003) to manage the number of components when performing probabilistic data association is developed. Iteratively merging terms of a Gaussian mixture using a pair-selection criteria based on the upper bound of the Kullback-Leibler divergence (Runnalls, 2007) between the reduced and original mixture was shown.

This paper proposes a formal, real-time GSF (RT-GSF) implementation using a novel combination of a GSF with Runnalls' condensation algorithm and an adaptive number of Gaussians using the MMSE estimate of the filter, to solve the fundamentally multi-modal problem of localizing an autonomous vehicle with

a known map in a sparse GPS environment. This paper empirically evaluates the performance and computation of the RT-GSF in general and the scheme for adjusting the number of terms in the posterior distribution more specifically. In addition, this paper statistically shows the GSF is able to better represent the posterior distribution of the vehicle pose when compared to the bootstrap particle filter. An analysis of the number of effective number of terms in the particle filter representation of the posterior distribution reveals a small term Gaussian mixture is capable of adequately describing the posterior distribution in real-time. The PF performance when the likelihood model and number of particles is changed is evaluated against the RT-GSF and the RT-GSF outperforms the PF regardless of location on the map due to the complex interaction of the measurement hypothesis tests. The computational load of the RT-GSF and PF is evaluated. The empirical evaluation uses experimental data recorded from the Cornell University autonomous vehicle (Miller et al., 2008a).

This paper is outlined as follows. Section 2.2 shows the RT-GSF and Runnalls' condensation technique. Section 2.3 describes the RT-GSF applied to the problem of map aided localization of the autonomous vehicle. Section 2.4 describes an experimental data collection with the Cornell University autonomous vehicle and metrics for comparing the RT-GSF and a PF implementation. Section 2.5 describes a detailed analysis of the RT-GSF with different number of terms in the posterior and the PF with different the number of particles. Finally, Section 2.6 summarizes with conclusions.

2.2 The Real-Time Gaussian Sum Filter

2.2.1 The Gaussian Sum Filter for Localization

The Bayesian estimation paradigm desires to determine the posterior density of the state x_k given a sequence of measurements $Z^k = \{z_0, \ldots, z_k\}$ from $t = 0$ to $t = t_k$. In the Gaussian Sum Filter (GSF), the posterior probability density $p(x_k|Z^k)$ is approximated as a sum of Gaussian densities (Anderson and Moore, 1979),(Alspach and Sorenson, 1972),(Sorenson and Alspach, 1971):

$$p(x_k|Z^k) \approx \sum_{i=1}^{N} \omega_k^i \cdot \mathcal{N}\left\{x_k; \hat{x}_k^i, P_k^i\right\} \tag{2.1}$$

where N is the number of components in the mixture, ω_k^i is the weight associated with the i^{th} Gaussian component, given as the multivariate normal density $\mathcal{N}\{x_k; \hat{x}_k^i, P_k^i\}$ with mean $\hat{x}_k^i$, and covariance P_k^i. The weights are constrained such that $\sum_{i=1}^{N} \omega^i = 1$. In the limit, as the number of Gaussians N approaches infinity and the covariance P_k^i goes to the zero matrix, the approximation in (2.1) can be used to represent any probability distribution with arbitrarily small statistical divergence (Sorenson and Alspach, 1971).

The discrete time state space system is modeled via nonlinear process and measurement functions:

$$x_k = f\left(x_{k-1}, u_{k-1}, v_{k-1}\right) \tag{2.2}$$

$$z_k = h\left(x_k, \mathcal{M}\right) + w_k \tag{2.3}$$

where in the process model (2.2), x_{k-1} is the state, u_{k-1} is the input to the system, v_{k-1} is the process noise, all at time t_{k-1}. In the measurement model (2.3), z_k is the measurement, x_k is the state and w_k is the measurement noise at time t_k. The

map-aided localization problem is then specifically considered by including $\mathcal{M}$ to represent the known map features.

The GSF uses the approximation that the d^{th} state transition density and m^{th} measurement likelihood are sums of Gaussian densities with N_{v_d} and N_{w_m} terms respectively, this formulation allows for d unique transition densities and m measurement functions:

$$p_d(x_k|x_{k-1},\mathcal{M}) \approx \sum_{j=1}^{N_{v_d}} \eta_{k-1}^{j} \cdot \mathcal{N}\left\{x_k; f^j\left(x_{k-1}, u_{k-1}, \hat{v}_{k-1}^{j}\right), Q_{k-1}^{j}\right\} \tag{2.4}$$

$$p_m(z_k|x_k,\mathcal{M}) \approx \sum_{l=1}^{N_{w_m}} \gamma_k^l \cdot \mathcal{N}\left\{z_k; h^l\left(x_k,\mathcal{M}\right) + \hat{w}_k^l, R_k^l\right\} \tag{2.5}$$

where the j^{th} term of the transition probability from time t_{k-1} to t_k is normally distributed with mean $f^j\left(x_{k-1}, u_{k-1}, \hat{v}_{k-1}^{j}\right)$ and covariance Q_{k-1}^{j} with component weight η_{k-1}^{j}. The known map information $\mathcal{M}$ is included for completeness, but the transition model is assumed independent of the known map, i.e. the vehicle is not constrained to lie on roads in the known map. In the measurement likelihood equation, the l^{th} component at time t_k is normally distributed with mean $h^l\left(x_k,\mathcal{M}\right) + \hat{w}_k^l$ and covariance R_k^l with component weight γ_k^l. Sorenson and Alspach (Sorenson and Alspach, 1971) assumed the process noise v_{k-1} and measurement noise w_k were Gaussian mixtures, but assumptions here generalize the entire state transition probability and measurement likelihood to be Gaussian mixtures. The added benefit of this representation is that the process model $f^j(\cdot)$ and measurement model $h^l(\cdot)$ are allowed to be different for each component, as are the means $\{\hat{v}^j, \hat{w}^l\}$ and covariances $\{Q^j, R^l\}$ of the process and sensor noises.

The filter starts at time t_{k-1} with an initial $N_{g_{k-1}}$ term Gaussian mixture representing the a priori probability density of the state x_{k-1}:

$$p(x_{k-1}|Z^{k-1}) \approx \sum_{i=1}^{N_{g_{k-1}}} \omega_{k-1}^{i} \cdot \mathcal{N}\left\{x_{k-1}; \hat{x}_{k-1}^{i}, P_{k-1}^{i}\right\} \tag{2.6}$$

where Z^{k-1} represents any a priori information in the system at initialization. The development here allows $N_{g_{k-1}}$, the number of components in the approximation of the posterior density, to vary as a function of time (Section 2.3.3). The prediction step of the GSF uses the EKF prediction step on each component of the prior mixture, although the Sigma Point Filter could be used. This gives a predicted density through the d^{th} transition density that is also a sum of Gaussian densities:

$$p(x_k|Z^{k-1}) \approx \sum_{q=1}^{N_{g_{k-1}} \cdot N_{v_d}} \bar{\omega}_k^q \cdot \mathcal{N}\left\{x_k; \bar{x}_k^q, \bar{P}_k^q\right\} \tag{2.7}$$

where $\{\bar{x}_k^q, \bar{P}_k^q\}$ is the one-step ahead predicted state and error covariance from the q^{th} EKF, where the index q iterates using the i^{th} prior component (2.6) and j^{th} transition probability component (2.4):

$$\bar{x}_k^q = f^j\left(\hat{x}_{k-1}^i, u_{k-1}, \hat{v}_{k-1}^j\right) \tag{2.8}$$

$$\bar{P}_k^q = F_{k-1}^q P_{k-1}^i F_{k-1}^{q^T} + \Gamma_{k-1}^q Q_{k-1}^j \Gamma_{k-1}^{q^T} \tag{2.9}$$

where F_{k-1}^q and Γ_{k-1}^q are the Jacobians of (2.2) with respect to the state x_{k-1} and process noise v_{k-1}, evaluated at the current state estimate, input, and process noise mean of the i^{th} state component mixture and the j^{th} noise component mixture:

$$F_{k-1}^q = \left.\frac{\partial f^j(x_{k-1}, u_{k-1}, v_{k-1})}{\partial x_{k-1}}\right|_{\hat{x}_{k-1}^i, u_{k-1}, \hat{v}_{k-1}^j} \tag{2.10}$$

$$\Gamma_{k-1}^q = \left.\frac{\partial f^j(x_{k-1}, u_{k-1}, v_{k-1})}{\partial v_{k-1}}\right|_{\hat{x}_{k-1}^i, u_{k-1}, \hat{v}_{k-1}^j} \tag{2.11}$$

Finally, the predicted weights on the Gaussian components are given by:

$$\bar{\omega}_k^q = \frac{1}{\bar{c}_k} \omega_{k-1}^i \eta_{k-1}^j \tag{2.12}$$

where $\bar{c}_k$ is a normalization constant to ensure $\sum_{q=1}^{N_{g_{k-1}} \cdot N_{v_d}} \bar{\omega}_k^q = 1$. For completeness, the indices in (2.7)-(2.12) are $i \in [1, N_{g_{k-1}}]$, $j \in [1, N_{v_d}]$ and $q = N_{v_d} \cdot (i-1) + j$.

The update step for the GSF follows similarly to the prediction step, where each component is updated with the measurements according to the EKF update step. The posterior density is approximated as:

$$p(x_k|Z^k) \approx \sum_{r=1}^{N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}} \omega_k^r \cdot \mathcal{N}\left\{x_k; \hat{x}_k^r, P_k^r\right\} \tag{2.13}$$

where $\{\hat{x}_k^r, P_k^r\}$ is the one-step ahead updated state and error covariance from the EKF with the m^{th} measurement, where the index r iterates using the q^{th} prior (2.7) and l^{th} measurement likelihood component (2.5):

$$\hat{x}_k^r = \bar{x}_k^r + K_k^r \left[z_k - h^l(\bar{x}_k^q, \mathcal{M})\right] \tag{2.14}$$

$$P_k^r = \left[I - K_k^r H_k^r\right] \bar{P}_k^q \tag{2.15}$$

and the Kalman gain K_k^r for the r^{th} filter is computed as:

$$K_k^r = \bar{P}_k^q H_k^{r^T} \left[H_k^r \bar{P}_k^q H_k^{r^T} + R_k^l\right]^{-1} \tag{2.16}$$

The matrix H_k^r in (2.15) and (2.16) is the Jacobian of the measurement function (2.3) with respect to the state x_k evaluated at the current best state estimate for the particular mixture component:

$$H_k^r = \left.\frac{\partial h^l(x_k^q, \mathcal{M})}{\partial x_k}\right|_{\bar{x}_k^q} \tag{2.17}$$

Lastly, the posterior component weight is computed with the measurement likelihood function as:

$$\omega_k^r = \frac{1}{c_k} \bar{\omega}_k^q \gamma_k^l \Lambda_k^r \tag{2.18}$$

where $\Lambda_k^r = p(z_k^l|x_k^q)$ is the likelihood function evaluated using the r^{th} predicted measurement and c_k is a normalization constraint to ensure $\sum_{r=1}^{N_{g_{k-1}} N_{v_d} N_{w_m}} \omega_k^r = 1$. For completeness, the indices in (2.13)-(2.18) are $q \in [1, N_{g_{k-1}} N_{v_d}]$, $l \in (1, N_{w_m})$ and $r = N_{w_m} \cdot (q-1) + l$.

2.2.2 Gaussian Mixture Condensation

The objective of condensation is to represent a Gaussian mixture with fewer components, but minimize the statistical difference between the full and reduced probability density representations. (Sorenson and Alspach, 1971) propose combining and eliminating components when the action induces an L^1 error below a given threshold. This is a form of truncation that does not preserve the overall mean and covariance of the entire mixture. Instead, (Salmond, 1988) proposes an iterative first and second moment preserving merging of components based on joining or clustering. The decision on which components to merge is based on an L^2-norm metric. Other methods that involve iterative merging use different metrics, including Williams' Integrated Square Difference similarity measure (Williams and Maybeck, 2003) and Runnalls' Kullback-Leibler Divergence Bound (Runnalls, 2007). Another technique proposed (Upcroft et al., 2004) draws random samples from the full-term Gaussian mixture and then uses Expectation-Maximization to fit a reduced term Gaussian mixture to the data points. Runnalls' joining algorithm is used here as the most accurate in terms of Kullback-Leibler divergence between the original and reduced mixture, preserves the first and second moments of the distribution, yet is computationally efficient for real-time mixture reduction during recursive estimation.

The moment preserving merge of any two weighted Gaussian densities $(\omega^i \cdot \mathcal{N}(x; \hat{x}^i, P^i))$ and $(\omega^j \cdot \mathcal{N}(x; \hat{x}^j, P^j))$ from a Gaussian mixture consists of computing a merged mean, covariance, and weight $(\omega^{ij} \cdot \mathcal{N}(x; \hat{x}^{ij}, P^{ij}))$ given that $\omega^i + \omega^j < 1$

(Williams and Maybeck, 2003):

$$\omega^{ij} = \omega^i + \omega^j \tag{2.19}$$

$$\hat{x}^{ij} = \frac{\omega^i}{\omega^{ij}}\hat{x}^i + \frac{\omega^j}{\omega^{ij}}\hat{x}^j \tag{2.20}$$

$$P^{ij} = \frac{\omega^i}{\omega^{ij}}P^i + \frac{\omega^j}{\omega^{ij}}P^j + \frac{\omega^i\omega^j}{(\omega^{ij})^2}\left(\hat{x}^i - \hat{x}^j\right)\left(\hat{x}^i - \hat{x}^j\right)^T \tag{2.21}$$

The condensation approach proposed by Runnalls' iteratively merges components based on a bounded approximation of the Kullback-Leibler divergence between the original and reduced mixture. The bound B^{ij} is given by:

$$B^{ij} = 0.5 \cdot \left[\left(\omega^i + \omega^j\right)\log\det(P^{ij}) - \omega^i \log\det(P^i) - \omega^j \log\det(P^j)\right] \tag{2.22}$$

where P^{ij} is the covariance from (2.21).

The RT-GSF proposed here performs the condensation algorithm at the end of each time step in the recursive RT-GSF. The goal is to reduce the number of mixands in (2.13) from $N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}$ back down to N_{g_k}. The merging is initialized by computing the bounded difference metric B^{ij} from (2.22) between each of the $N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}$ components in the posterior mixture (2.13). Then, the two components with the smallest metric are merged according to (2.19)-(2.21). The routine continues by computing the difference metric B^{ij} between the new merged component and the remaining components, and then merging components with the smallest B^{ij} until the desired number of terms (i.e. N_{g_k}) is reached. The computational requirements are $O(N^2)$, where N is the number of terms in the Gaussian mixture prior to condensation. In addition, a time-varying desired number of terms is included in this formulation; the desired number of terms is adapted depending on the estimate uncertainty.

2.3 Real-Time GSF Application to Map Aided Localization

2.3.1 Vehicle Prediction

The vehicle state $x_k = [e_k; n_k; h_k]$ is defined with respect to the map center, where e_k and n_k are the position in the map, and h_k is the heading. The RT-GSF requires predicting the previous posterior density (2.6) forward to generate the a priori density (2.7) for measurement fusion. The single transition probability density (2.4) is well represented with a single Gaussian density, $N_{v_{\text{motion}}} = 1$, because the high update rate of the odometry information results in very small nonlinearities during state prediction and $d = 1$, because this process model represents the vehicle for all motion. A fourth element is added to the vehicle state to allow prediction using a 4×4 rotation and translation matrix:

$$\begin{bmatrix} e_k \\ n_k \\ h_k \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(h_{k-1}) & -\sin(h_{k-1}) & 0 & e_{k-1} \\ \sin(h_{k-1}) & \cos(h_{k-1}) & 0 & n_{k-1} \\ 0 & 0 & 1 & h_{k-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{\nu}_{k-1} \tag{2.23}$$

where $\hat{\nu}_{k-1} = [\hat{\nu}_{x_{k-1}}, \hat{\nu}_{y_{k-1}}, \hat{\nu}_{h_{k-1}}, 1]^T$ is the measured incremental position and heading input change in the vehicle coordinate system (front, drive door, roof) measured from the odometry system (INS) as a summation of the true system input u_{k-1} and the process noise v_{k-1}. Each component in the Gaussian mixture is predicted forward using (2.23) and the state covariance is computed according to (2.9). The Jacobian of (2.23) with respect to the state x_k and the process noise v_k is actually 4×4, but only the upper left 3×3 block is used in the covariance propagation.

2.3.2 Measurement Update

The $N_{g_{k-1}}$ term predicted density from (2.7) is updated with absolute, stopline and lane offset measurements using (2.14)-(2.18) to yield the posterior density (2.13) as a Gaussian mixture. This process expands the number of components in the Gaussian mixture to $N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}$. There are $m = 4$ different types of measurements; absolute position (a), stopline (s) and two lane-offset (o) measurements that are described below. Each with a number N_{w_ϑ} of component Gaussians, where $\vartheta \in \{a, s, o\}$. In the implementation discussed here, the high rate odometry data is used as a time synchronizer. The RT-GSF is allowed to run asynchronously between odometry measurements and expands the number of Gaussians to as many components as are successfully updated in between odometry updates, but before each new odometry update condensation is performed. This allows a variable number of terms to represent the posterior density before condensation, but a fixed number after condensation.

Absolute Position Measurements

The tightly coupled GPS / INS system provides absolute position measurements z_{a_k} that are used to update the vehicle state estimate. The measurement likelihood for the absolute position measurements is well modeled as a single Gaussian given a component of the predicted Gaussian mixture:

$$p(z_{a_k}|\bar{x}_k^q) \sim \mathcal{N}\left\{z_{a_k}; \bar{x}_k^q, R_{aa_k}\right\} \tag{2.24}$$

where R_{aa_k} is the covariance of the absolute position measurement at time t_k. Unfortunately, as in (Miller and Campbell, 2008), the absolute position estimates generated via the recursive information filter are correlated from one time step

to the next, because they are the output of a model based filter. To account for this autocorrelation, the measurements are whitened by augmenting the state with East and North GPS biases $\beta_k^q = [\beta_{e_k}^q, \beta_{n_k}^q]$ that have the following dynamics:

$$\beta_k^q = \lambda \beta_{k-1}^q + v_{\beta_{k-1}}^q \tag{2.25}$$

where $\lambda = \exp(-\Delta T/T_b)$ accounts for the bias's autocorrelation time T_b during the time interval from t_{k-1} to t_k, and $v_{\beta_{k-1}}^q \sim \mathcal{N}(0, Q_{\beta\beta})$. The addition of the bias terms now enables the assumption that the likelihood of the absolute position measurements is Gaussian and white:

$$p(z_{a_k}|\bar{x}_k^q, \beta_k^q) \sim \mathcal{N}\{z_{a_k}; \bar{x}_k^q + \beta_k^q, R_{aa_k}\} \tag{2.26}$$

The absolute position measurement is used to update each component of the predicted Gaussian mixture according to (2.14)-(2.18).

Map Relative Position Measurements: Stopline and Multi-Modal Lane Offset

Unlike absolute position measurements, relative position measurements are not straightforward projections of the vehicle position. Instead, the measurements are accurate ranges to nearby stoplines or distances from lane boundaries relative to the vehicle. The relative measurements are combined with the known map to generate weak absolute position information that is fused into the global state estimate. The relative measurements generated from the vehicle cameras allow the vehicle to maintain a global estimate in the absence of absolute position measurements.

The first relative position measurement comes from the stopline camera that detects the range from the vehicle to a stopline in the camera field-of-view. The

measurement likelihood for the stopline detection is given by a single Gaussian as:

$$p(z_{s_k}|\bar{x}_k^q, \mathcal{M}) \sim \mathcal{N}(z_{s_k}; \bar{z}_{s_k}^q, R_{ss_k}) \tag{2.27}$$

where $\bar{z}_{s_k}^q$ is the Euclidean distance from the q^{th} component of the predicted density to the nearest stopline in the map $\mathcal{M}$.

The second type of relative position measurement is generated from the lane finding system that detects bounding lines in the vision image. The measurement z_{o_k} is the perpendicular distance from the detected lane boundaries and the camera heading with respect to the occupied lane. The challenge in incorporating the lane offset measurement is that the vision processing algorithms generate errors that are not well-modeled with a single Gaussian. Manual analysis of the vision processing algorithm revealed the errors occur depending on the modality of the lane identification. Therefore, the measurement likelihood function is best represented as sum of Gaussians, making the RT-GSF an ideal filter for these measurements. The different components of the measurement likelihood come from the different lane detection modes that arise in the vision processing algorithm. There are $N_{w_{\text{Lane}}} = 6$ different modes of detection by the lane-finding algorithm: detecting the correct lane; detecting the lane to the left of the true lane; detecting the lane to the right of the true lane; the combination of the left and correct lane; the combination of the right and correct lane, and the sixth is the combination of the left, correct, and right lane (Miller et al., 2011). The key insight with these uncertainties is that when conditioned on the specific detection mode, the measurement likelihood is well-modeled as a single Gaussian:

$$p(z_{o_k}|\bar{x}_k^q, \mu_l, \mathcal{M}) \sim \mathcal{N}\{z_{o_k}; \bar{z}_k^r, R_{oo_k}^l\} \tag{2.28}$$

where μ_l indicates the l^{th} measurement mode, and $\bar{z}_k^r$ is the predicted measurement with respect to the map $\mathcal{M}$ from the q^{th} component and l^{th} detection mode (*recall*:

$r = (N_{w_m} \cdot q + l)$). The overall measurement likelihood function for the vision system is then given by the Gaussian mixture:

$$p(z_{o_k}|x_k^q, \mathcal{M}) \sim \sum_{l=1}^{N_{w_{\text{Lane}}}=6} \gamma_k^l \cdot \mathcal{N}\{z_{o_k}; \bar{z}_k^r, R_{oo_k}^l\} \tag{2.29}$$

2.3.3 Adaptive Condensation Components

A key element of the GSF is the condensation of components in the posterior distribution after each time step. For a static update, where the same update occurs at each time step, the number of components in the posterior density after condensation must be less than or equal (on average) to the number of components in the original mixture before prediction to avoid a geometrically growing number of Gaussians, or $N_{g_k} \leq N_{g_{k-1}}$. The formulation presented here however, considers an adaptive update of the number of components.

The number of components in the posterior density after condensation at time t_k, given the number of components at t_{k-1} is adjusted according to:

$$\begin{aligned}
N_{g_k} &= N_{g_{k-1}} + 1, \text{ if : } \; tr[P_{k_{\text{MMSE}}}] > \tau_{\text{up}}, \; N_{g_{k-1}} < N_{g_{\max}} \\
N_{g_k} &= N_{g_{k-1}} - 1, \text{ if : } \; tr[P_{k_{\text{MMSE}}}] < \tau_{\text{dn}}, \; N_{g_{k-1}} > N_{g_{\min}} \\
N_{g_k} &= N_{g_{k-1}}, \qquad \text{otherwise}
\end{aligned} \tag{2.30}$$

where N_{g_k} is the number of components after condensation at time t_k, and $N_{g_{k-1}}$ is the number components in the posterior at the previous time step. The thresholds are τ_{up} and τ_{dn}, and $N_{g_{\max}}$ and $N_{g_{\min}}$ are integer numbers for the maximum and minimum number of components allowed to represent the posterior. The specific values for τ_{up} and τ_{dn} were determined using a cross-validation approach. The values were tuned on half the dataset and evaluated using the other half of the dataset.

This enables the number of Gaussians to expand/contract based on the needs of the filter (i.e. representing the posterior), yet caps the number of Gaussians so the filter does not grow without bound.

The number of components should be enough to accurately represent a multi-modal density, but not too many such that they are representing the same information. In this light, the number of components after condensation is increased or decreased based on the trace of the MMSE covariance matrix $P_{k_{\text{MMSE}}}$ computed from the entire Gaussian mixture according to (2.32). The trace $tr[P_{k_{\text{MMSE}}}]$ is used because it is fast to compute and indicates increased variance requiring additional mixands or reduced variance requiring fewer mixands.

To generate a single estimate at the end of the predict-update cycle, which is usually required for planning in autonomous systems, one approach is to compute the MMSE estimate. The MMSE estimate is computed for the state and state error covariance as a metric for evaluation, although it is not necessary for the RT-GSF itself:

$$\hat{x}_{k_{\text{MMSE}}} = \sum_{r=1}^{N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}} \omega_k^r \cdot \hat{x}_k^r \tag{2.31}$$

$$P_{k_{\text{MMSE}}} = \sum_{r=1}^{N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}} \omega_k^r \left[P_k^r + (\hat{x}_k^r - \hat{x}_{k_{\text{MMSE}}})(\hat{x}_k^r - \hat{x}_{k_{\text{MMSE}}})^T \right] \tag{2.32}$$

This completes one cycle (predict, update) of the RT-GSF.

2.3.4 Statistical Testing

In real-time implementations, it is necessary to perform statistical testing on measurements prior to incorporating into the posterior to determine if they are statistically viable (Bar-Shalom et al., 2001). This avoids problems of divergence in

a real-time filter due to spurious measurements that are not well-modeled by the sensor likelihoods. For example, in the urban environment, multipath on the GPS signals may cause erroneous position measurements to be fused into the state. To prevent any Gaussian sum component from being updated with an erroneous measurement, a probabilistic hypothesis test is performed on each component and mode of the measurement likelihood before the mixture component is updated. First, an innovation statistic is computed for the r^{th} mixture component in the predicted density (2.7):

$$\epsilon^r_{\vartheta_k} = \left(z_{\vartheta_k} - \bar{z}^r_{\vartheta_k}\right)\left(S^r_{\vartheta_k}\right)^{-1}\left(z_{\vartheta_k} - \bar{z}^q_{\vartheta_k}\right)^T \tag{2.33}$$

where $\bar{z}^r_{\vartheta_k}$ is the mode-conditioned predicted measurement, $S^r_{\vartheta_k}$ is the innovation covariance (bracketed term in (2.16)) and $\vartheta \in \{a, s, o\}$ is each measurement type. The innovation statistic for each component is distributed according to a χ^2 random variable with n_z degrees of freedom (Bar-Shalom et al., 2001), where n_z is the cardinality of the measurement space. The threshold is set at the 95% bound of $\chi^2_{n_z}$ random variable and tests if the measurement was likely to have come from the predicted state. The absolute position, stopline and lane offset measurements have 3, 1, and 2 degrees of freedom respectively. If the hypothesis test fails, the measurement is not fused into the vehicle state estimate, and the algorithm continues to the next component.

The probabilistic hypothesis test is performed for each component and each likelihood mode. This is an important distinction for the RT-GSF proposed here, because each mixand of the prior density (2.7) can be independently pairwise evaluated according to the mode of the predicted distribution and likelihood function. Ideally, the number of updated states moves from $N_{g_{k-1}} \cdot N_{v_d}$ to $N_{g_{k-1}} \cdot N_{v_d} \cdot N_{w_m}$, but not all measurements are fused with each component. Instead, only the mode-conditioned measurements that pass the hypothesis test are updated. If

a measurement-mode and component pair are rejected, that is equivalent to assigning the weight in (2.18) to zero. If all of the measurement detection modes are rejected for a given component, that component is not updated and the measurement is rejected entirely.

2.4 Experiment and Metrics for Comparison

A field experiment was conducted to demonstrate the Real-Time GSF as applied to the multi-modal localization problem. Sensor-data was collected in order to systematically evaluate and study the RT-GSF with respect to a truth measure. The RT-GSF is evaluated with different values on the minimum and maximum number of terms in the posterior after condensation. In addition, a systematic comparison is made with the bootstrap PF implementation (Miller and Campbell, 2008) in terms of both performance and computation. Two different weighting functions on the importance samples and the number of particles in the PF was varied to quantify the differences between the RT-GSF and PF. The posterior distribution representation of the PF is evaluated offline by computing an effective number of Gaussians which enables direct comparison with the RT-GSF.

2.4.1 Autonomous Testbed

Experimental data was collected to analyze the RT-GSF using Cornell University's autonomous Chevrolet Tahoe (Miller et al., 2008a), shown in Figure 2.1. The autonomous Tahoe is equipped with a Septentrio PolaRx2e@ tri-antenna GPS receiver providing GPS position, velocity, and attitude, a Northrop Grumman LN-

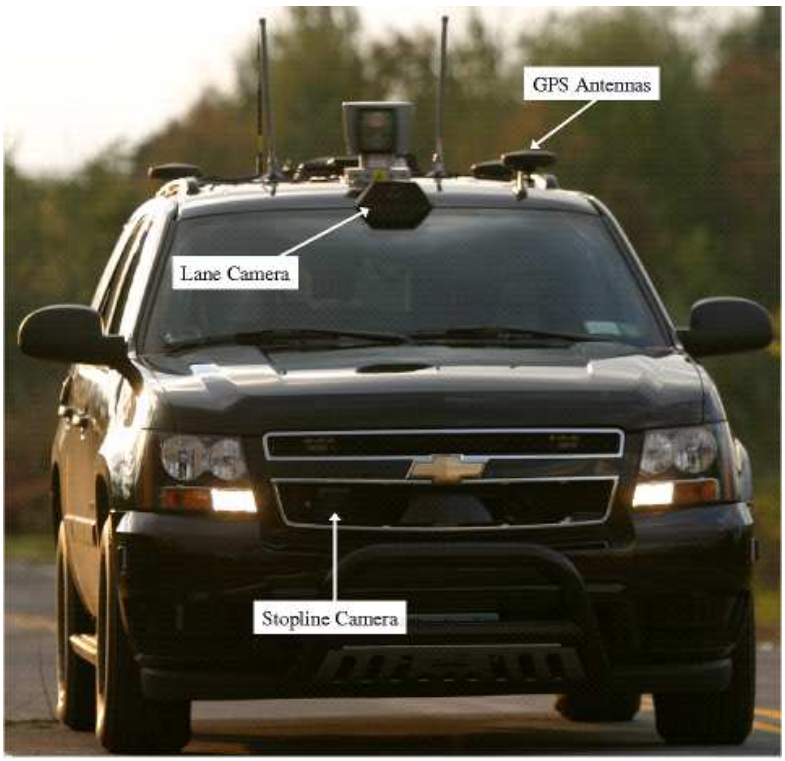

Figure 2.1: Cornell University's autonomous Chevrolet Tahoe, equipped with GPS, inertial navigation, and vision-based lane sensing, line sensing, and stopline detection. The Tahoe is shown here on Cornell University? autonomous vehicle test course at the Seneca Army Depot in Romulus, NY, where data collection for these experiments was performed.

200 tactical grade inertial measurement unit (IMU) and anti-lock brake (ABS) wheel encoders. The GPS antennas are mounted on the Tahoe's roof, and they are tightly coupled with the IMU and wheel odometry in an Extended Square Root Information Filter to produce the filtered GPS/INS navigation solution (Miller et al., 2008b). This filter provides PosteriorPose with absolute position measurements.

The Tahoe is also equipped with a Basler A622F camera, mounted in the center of the roof, running MobilEye SeeQ software and vision module providing road lane and line estimates, and Cornell's own texture and edge-based vision algorithms providing redundant road and lane line estimates. The Tahoe also has

a Basler A311F camera, mounted in the Tahoe's front bumper, with a Cornell stopline detection system (Miller et al., 2008a).

The tightly-coupled GPS / INS solution provides vehicle position and odometry measurements at 100 Hz. The GPS/INS navigation solution, including fused data from the vehicle wheel encoders, is provided in the absence of GPS signals and when GPS signal is absent is referred to as the 'integrated INS solution.' The MobilEye runs at $\approx$ 18 Hz, Cornell's lane detection system runs at $\approx$ 3 Hz, and the stopline algorithm runs at $\approx$ 20 Hz.

2.4.2 Experimental Data Collection

The RT-GSF as applied to the localization problem was evaluated using Cornell University's autonomous vehicle driving on a vehicle test course. The test course consists of several miles of accurately surveyed roads, including road segments with accurately painted road lines / stoplines as well as segments with no road lines; an overhead view of the course is shown in Figure 2.2. In addition, there are features on the course that are not included in the known map, including additional painted stoplines and railroad tracks. Experimental data was collected for 37 minutes in an area with a clear view of the sky which provided continuous collection of OmniSTAR high precision differential (HP) corrections signal, accurate to 10 cm; this signal is used as ground truth, and neither the PF nor RT-GSF has access to this signal after initialization. The tests were performed driving at city speeds up to 30 mph (13.4 m/s). The sensor data was collected and logged for post-processing to evaluate different filter configurations offline. Each filter uses the known map and observations of the features in the map to perform global localization. The estimator is not constrained to have the solution lie on the known map, however the

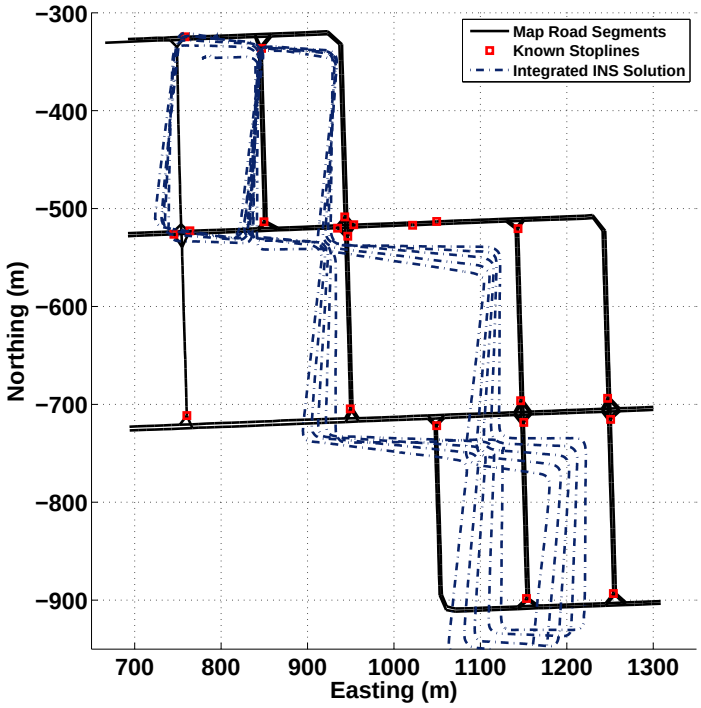

Figure 2.2: Overhead view of a portion of Cornell's autonomous test site, which consists of road segments and stoplines. An integrated inertial navigation solution overlaid on the map shows the degradation of vehicle pose estimates in the absence of additional localization information such as GPS or map information.

observability of features from the map is required for consistent global localization.

Figure 2.2 shows the integrated INS solution overlaid on the road network. The INS solution diverges significantly from the road during the artificial GPS blackout, since there are no absolute position measurements to correct numerical integration of small incremental acceleration and angular velocity errors. A further discussion of INS performance is in Section 2.5.1. This demonstrates the need for relative position measurements to maintain accurate global positioning in an extended GPS blackout.

2.4.3 Filter Localization Accuracy Performance

To evaluate the performance of the various filters for localization, the filter estimate is compared against truth data. The truth data is the east and north position of the vehicle. The truth data is obtained from the High Precision GPS signal localization estimate available during the experiment. The High Precision solution has an absolute error less than 0.10 m. The filter estimates are compared against the truth data using position errors at a rate of 10 Hz according to the RMS of the position errors:

$$E_k^{\text{Filter}} = \left|\left|[e_k^{\text{HP}}, n_k^{\text{HP}}]^T - [\hat{e}_k^{\text{Filter}}, \hat{n}_k^{\text{Filter}}]^T\right|\right| \tag{2.34}$$

where 'HP' denotes the High Precision signal taken as truth and 'Filter' is the MMSE estimate of the algorithm being evaluated. The heading portion of the state is discarded for statistical analysis as it did not have a statistically significant relationship.

2.4.4 Estimate Effective Number of Gaussians in a PF for Comparison

A key contribution of this work is to systematically compare the RT-GSF with the bootstrap PF (sections 2.5.3 - 2.5.8); in order to do that, however, a common representation and performance metric is required. Following condensation the RT-GSF approximates the posterior density as a weighted sum of Gaussians. However, the PF implementation represents the posterior density as a discrete set of weighted samples. In an effort to evaluate the accuracy and applicability of the RT-GSF and

make direct comparison with the PF, an off-line fitting procedure is run where the samples from the PosteriorPose PF (Miller and Campbell, 2008) at each time step are fit to a Gaussian mixture using the Expectation-Maximization (EM) algorithm (Bishop, 2006). The EM algorithm is initialized using the K-Means clustering technique (Bishop, 2006) and is used along with the Bayesian Information Criterion (BIC) (Bishop, 2006) to determine the effective number of Gaussians in the PF posterior samples. This idea of fitting the PF data to a Gaussian mixture is similar to the Gaussian Particle Filter (Kotecha and Djuric, 2003a) and Gaussian Sum Particle Filter (Kotecha and Djuric, 2003b), where the particles at each time step are approximated with a Gaussian distribution that is used for resampling.

The EM and K-Means algorithms require performing the Gaussian mixture fit with a set number of mixture components. To determine the number of terms required to represent the particle filter posterior density, the EM algorithm is run repeatedly with an increasing number of components M in the Gaussian mixture. Then, the Bayesian Information Criterion (BIC) (Bishop, 2006) is used to determine the M-term Gaussian mixture that best fits the PF data:

$$\text{BIC} \approx -2 \ln p(X|\mu, \Sigma, \pi) + \frac{1}{2} M \ln N \tag{2.35}$$

where $\ln p(X|\mu, P, \omega)$ is the log-likelihood from the EM algorithm (Bishop, 2006), M is the number of free parameters, and N is the number of data points. The number of components in the Gaussian mixture with the minimum BIC is the effective number of Gaussians for the particle filter posterior density. Note that the BIC penalizes overly complex models more heavily than the Akaike Information Criterion (AIC) (Bishop, 2006) and for this reason was chosen as the metric to determine the effective number of Gaussians.

2.5 Experiment Results and Analysis

2.5.1 RT-GSF Localization Performance

To demonstrate the localization accuracy when using vision based relative landmark measurements with the RT-GSF, the RT-GSF was run over the same dataset as the integrated INS solution shown in Figure 2.2 in order to estimate the global east and north positions. These estimates are compared against the truth data according to (2.34).

After a 5 minute stationary initialization period during which GPS measurements are available, the RT-GSF then estimates the vehicle pose while no GPS signal is present. In the first case presented here, the number of Gaussians after each cycle collapses back to $N_{g_{\min}} = N_{g_{\max}} = 5$. An additional analysis varying the number of mixands in the posterior after condensation is described in Section 2.5.2.

Figure 2.3 shows the RMS position error of the RT-GSF and integrated INS solution over the entire course including a 32 minute extended GPS blackout. Notice that the INS solution (also shown in Figure 2.2) diverges as the vehicle travels from the initial position, leading to errors of up to 114 m. The INS error oscillates, because of a bias error in the heading. The errors is maximized when the vehicle is furthest from the point of the initial blackout and minimized when the vehicle returns to its initial configuration. The peak error is reduced due to drift in the error bias, but if the GPS blackout were to persist, the integrated error would grow without bound. The RT-GSF, however, remains converged and the average error is 1.35 m, standard deviation is 0.90 m and the maximum error is 4.74 m,

which occurs after maneuvering through an intersection. This demonstrates the RT-GSF is able to solve the fundamentally multi-modal localization problem.

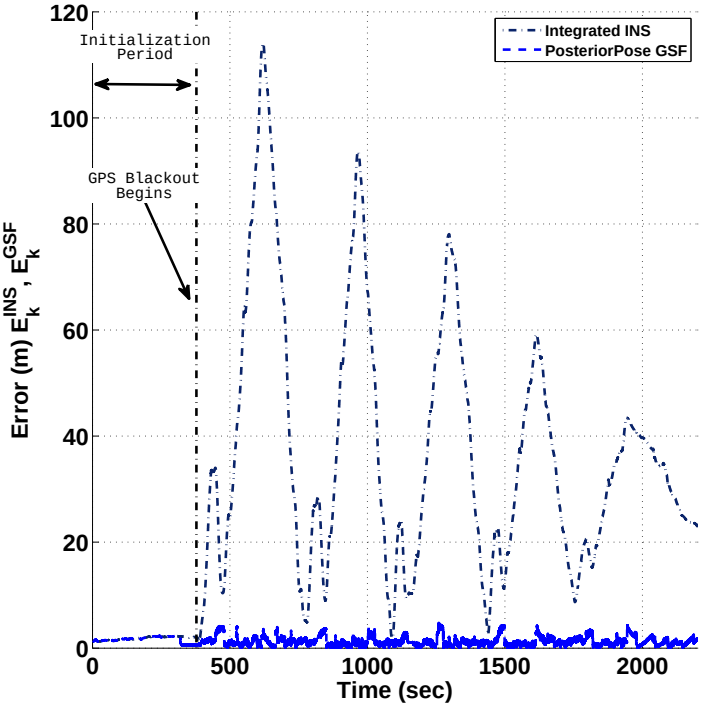

Figure 2.3: Plot of the position error over the entire data collection from the INS solution and RT-GSF. RT-GSF uses weak localization information provided by vision-based camera measurements of lane lines and stoplines to remain converged over a 32 minute GPS blackout. The INS solution, which does not use map information, diverges due to incorrect inertial estimates that cannot be corrected after losing GPS signals.

2.5.2 Varying the Number of Mixands in the RT-GSF After Condensation

To investigate the impact of varying the number of mixands in the RF-GSF after condensation, a trade study was performed that varied $N_{g_{\max}} \in [1, 15]$ for $N_{g_{\min}} \in \{1, 2, 5\}$ from equation (2.30). The objective is to demonstrate the need to maintain a small number of term multi-modal posterior distribution in order to enable convergence of the localization solution. Figure 2.4 shows the average RMS position error of the RT-GSF MMSE estimate compared to the 'HP' truth data. The average error is shown along with the standard deviation as a function of the maximum number of mixands $N_{g_{\max}}$ (horizontal axis). In addition, the study is performed using three different values of the minimum number of mixands $N_{g_{\min}}$ (different color bars). The maximum error when $N_{g_{\min}} = 1$ was at least 18 m regardless of the value for $N_{g_{\max}}$. In addition, the performance for $N_{g_{\min}} = \{2, 5\}$ shows little variation regardless of the value of $N_{g_{\max}}$. This demonstrates it is necessary to maintain a small term $N_{g_{\min}} \geq 2$ multi-modal posterior distribution to ensure capture of the true data and the best performance.

2.5.3 Comparison of Performance Between RT-GSF and PF

The RT-GSF's state estimates are compared to the PF PosteriorPose (Miller and Campbell, 2008) algorithm using 2000 particles run on the same data. The same measurement likelihood functions are used along with the same hypothesis tests for each case, although the outcome of the hypothesis tests can be different. The

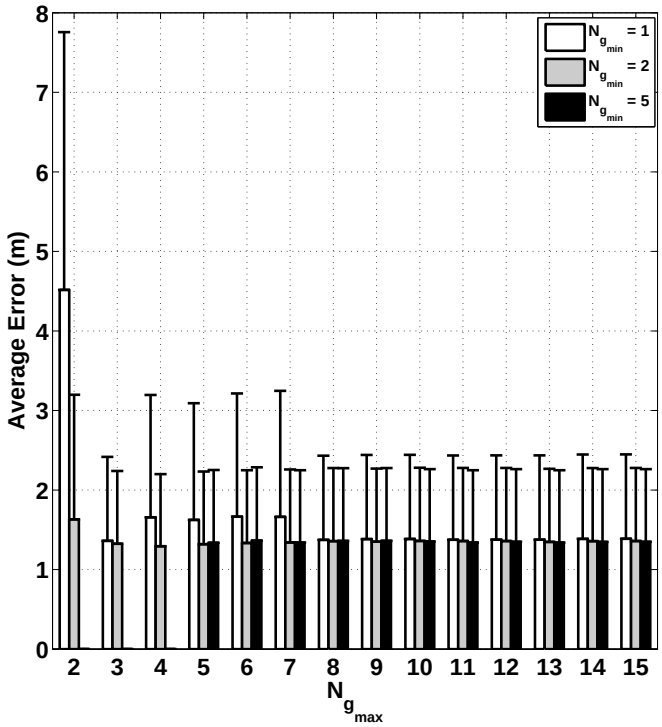

Figure 2.4: RT-GSF average position error over the entire data collection for different values of $N_{g_{\min}}$ and $N_{g_{\max}}$. The best performance occurs with at least 2 terms in the posterior and increasing the number of terms has similar performance.

RMS position error for each of the algorithms is shown over the entire 37 minute run in Figure 2.5. There are times where each algorithm performs better than the other. The PosteriorPose PF algorithm average error is 1.85 m (compared to 1.3 m for the RT-GSF), the standard deviation is 1.6 m and the maximum error is 11.3 m, after exiting a turn. This shows both algorithms are able to remain converged in the presence of an extended GPS blackout. Figure 2.5 shows the PF diverges shortly before and after the GPS blackout. If this error is excluded, the average error is 1.63 m, the standard deviation is 1.3 m and the maximum error is 5.6 m. Excluding this region does not change the conclusions for the remainder of the paper and is included in the remaining analysis.

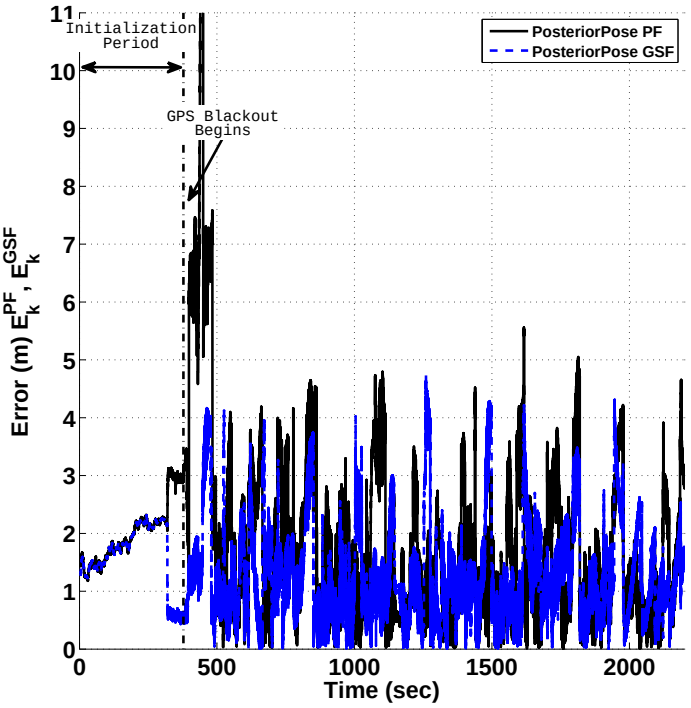

Figure 2.5: RT-GSF and particle filter position errors over the entire data collection, including the GPS blackout. Both solutions remain converged over the entire 32 minute blackout.

A more direct comparison can be made by subtracting the PF error at each time step E_k^{PF} from the RT-GSF error at each time step E_k^{GSF} to create a difference in error $\Delta_k = E_k^{\text{PF}} - E_k^{\text{GSF}}$. The histogram of the difference in the error between the RT-GSF and PF is shown in Figure 2.6. On average, over the entire run, the RT-GSF performs $\bar{\Delta} = 0.52$ m better than the PF. When a paired t-test is performed, the 0.52 m average performance improvement of the RT-GSF is statistically significant at the $p = 0.05$ level. In fact, the t-test concludes the RT-GSF outperforms the PF over the 37 minute run by at least 0.51 m at the 5% significance level. The histogram in Figure 2.6 has a sub-mode around 2.5 m. This indicates the RT-GSF is 2.5 m closer than the PF to the true position. This occurs when the PF has converged to the incorrect lane and the RT-GSF converges to

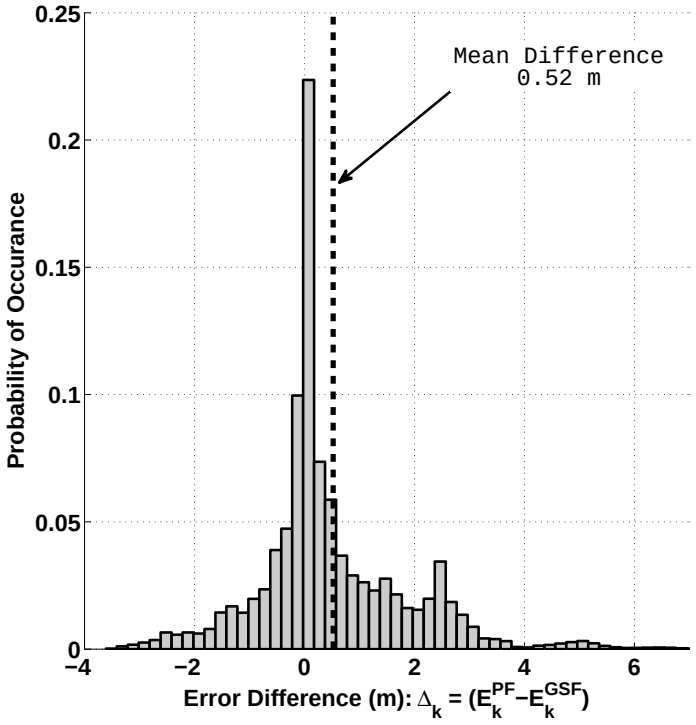

Figure 2.6: Histogram of the difference in the position error (Δ_k) between the RT-GSF and PF. The RT-GSF outperforms the PF with statistical significance.

the correct lane.

An interesting and insightful area of comparison between the RT-GSF and PF is when the autonomous vehicle approaches and completes a 90° right turn. This is a particularly challenging case for localization because the map information is sparse and the lane of the new road segment the vehicle occupies is uncertain. Figure 2.7 is an example of this case and shows the RMS position error of the RT-GSF and the PF 1054 seconds into the data capture (676 seconds after GPS blackout). Figure 2.7 shows the vehicle approach a corner and after maneuvering through the corner, the PF converges to the incorrect lane. The PF remains converged to the incorrect lane until encountering a stop sign 58 seconds later.

The RT-GSF, however remains converged to the correct lane the entire time. This incorrect convergence is the primary reasons the RT-GSF is able to statistically out-perform the PF.

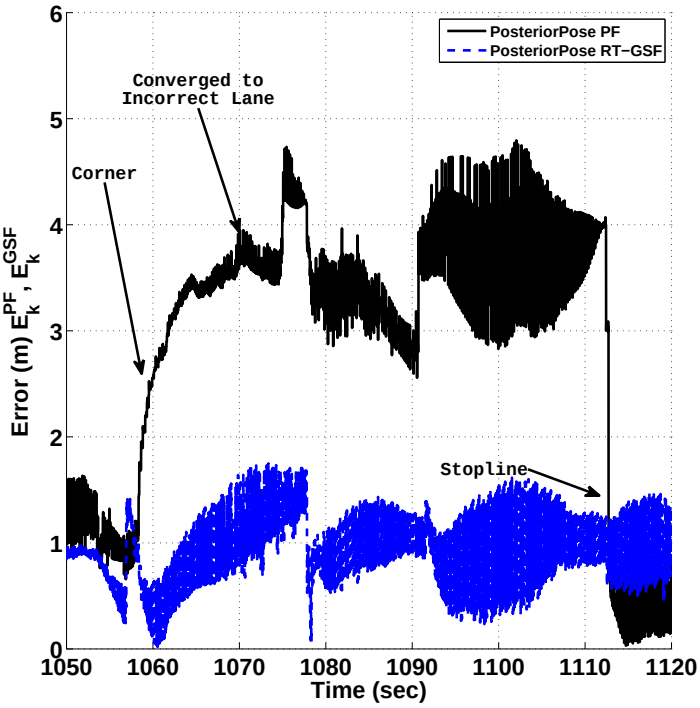

Figure 2.7: Plot of the position error from the RT-GSF and PF over a subset of the data collection. The autonomous vehicle makes a right turn and the PF converges to the incorrect lane, while the RT-GSF is able to converge to the correct lane. The PF remains diverged until observing a stopline at 1112 seconds.

Figure 2.8 depicts the RT-GSF (left) and PF (right) posterior density approximations at three time snapshots just before (first row), during (second row) and after (third row) encountering the corner; this corresponds to the beginning of Figure 2.7. Figure 2.8 (first row) shows a nearly unimodal and consistent posterior distribution prior to the corner for the RT-GSF and PF. Figure 2.8 (second row) shows the PF converges to two distinct clusters in the turn, but the RT-GSF,

using $N_g = 5$ terms in the Gaussian mixture, overlaps both these distinct modes. Figure 2.8 (third row) shows the estimated vehicle pose posterior three seconds after the turn. The RT-GSF MMSE estimate converges to the correct lane and the PF converges to the incorrect lane. As a result of this better representation during the turn, the RT-GSF converges to the correct lane three seconds after the vehicle has moved through the corner. Thus, because the PF fails to maintain an accurate representation of the posterior density as the vehicle moves through the corner, the PF position estimate converges to the incorrect lane after the turn. The sequence of frames in Figure 2.8 demonstrates the difficulty that the PF has in representing the true posterior density in complex scenarios.

2.5.4 Effective Number of Gaussians

It is insightful to interpret the PF with a Gaussian mixture representation in order to understand how the effective number of Gaussians that match the PF data varies over the data run. The effective number of Gaussians of the PF is calculated following the description of the EM algorithm in Section 2.4.4; this process is run offline, over the stored PF data at each time step in order to estimate the effective number of Gaussians in the posterior distribution. The EM algorithm is run 100 times with different initial means and covariances for each of the K-terms in the Gaussian mixture to encourage convergence to a global minimum. Each iteration is allowed to run to convergence for each M-term Gaussian mixture, where $M \in [1, 15]$. The number of occurrences for each M-term Gaussian mixture representation of the PF is shown in Figure 2.9. The average effective number of Gaussian distributions in the PF posterior density is 2 and fewer than 5 components for the majority of the data, despite the particle filter using 2000 particles. This

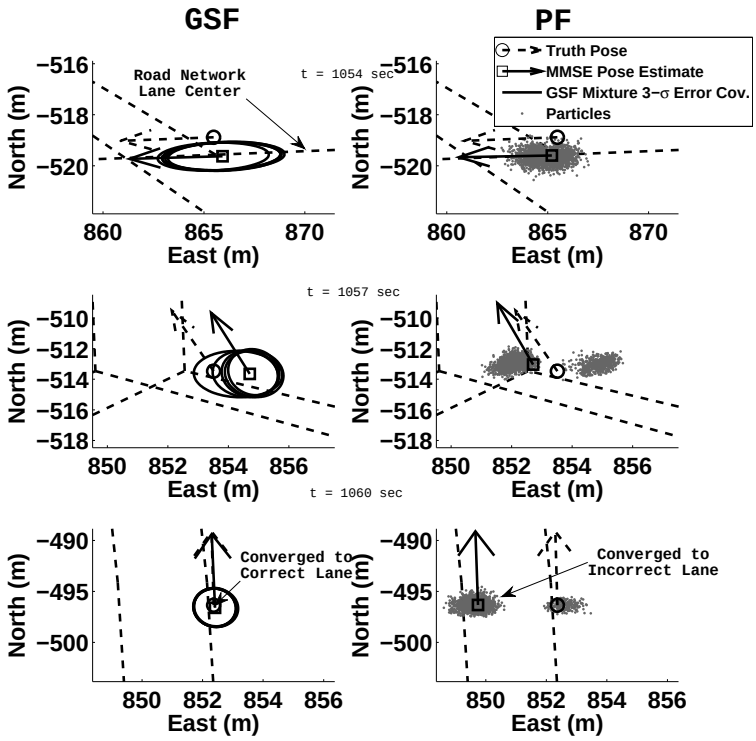

Figure 2.8: Sequence of frames beginning 1054 seconds into the experiment, showing the autonomous vehicle maneuvering through a corner, comparing the performance and behavior between the RT-GSF and PF. The PF converges to the incorrect lane, while the RT-GSF converges to the correct lane after the corner, because the PF posterior distribution does not represent the true errors.

suggests the PF is inefficient for this application; it uses 2000 particles to represent a posterior distribution equivalent to a Gaussian mixture with as few as 2 terms.

Figure 2.10 shows the areas on the map where the RT-GSF adaptively increases the number of terms in the posterior distribution prior to condensation. For this analysis, the case where $N_{g_{\min}} = 1$ and $N_{g_{\max}} = 8$ was used. Many of the occurrences are around corners, but of particular interest is the area of the map with the poorest lane markings in the lower right portion of the course that shows the RT-GSF increasing the number of terms in the posterior. This is caused, because

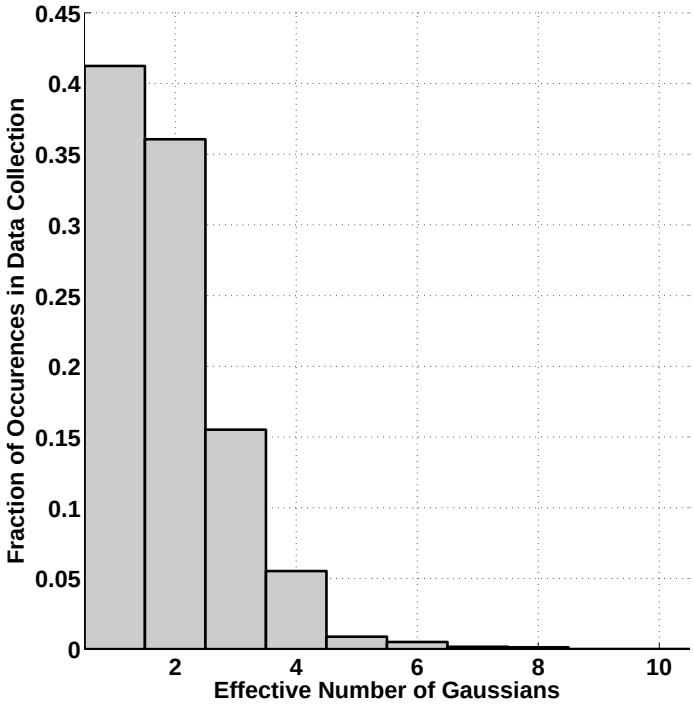

Figure 2.9: Histogram showing the effective number of Gaussian components in the PF data over the entire 37 minute data collection. The PF uses 2000 particles and a M-term Gaussian mixture is fit to the data. The majority of the PF data is well represented with fewer than 5 Gaussians.

the uncertainty in the MMSE estimate grows as few quality vision measurements are fused. This allows the RT-GSF to adaptively increase the number of terms in the posterior to account for the added uncertainty.

In order to examine when the PF has the most multi-modal content, and thus the most challenging areas, the points where the effective number of Gaussians is greater than 4 are extracted, and plotted with the true vehicle position on the test course in Figure 2.10. The majority of the complex PF posterior distributions occur around intersections, which is where the vehicle has the most opportunity to predict incorrect measurement-feature data assignments.

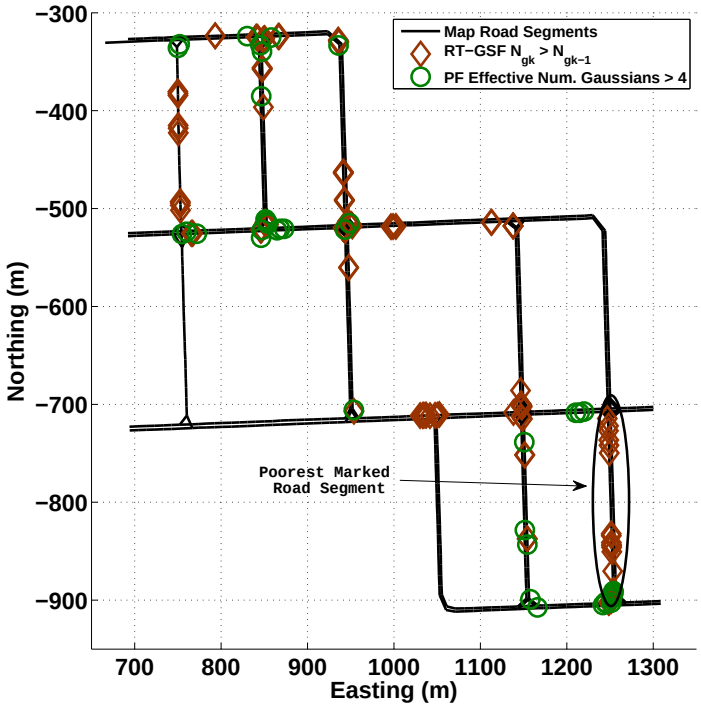

Figure 2.10: Plot of location where adaptation occurs in the RT-GSF ($N_{g_{\min}} = 1$, $N_{g_{\max}} = 8$) and the most complex PF posterior densities. The majority of the points occur around intersections on the map or areas of poor lane markings. The RT-GSF adapts the number of terms in the posterior based on uncertainty caused from poor lane-markings and data association error. The PF posterior complexity increases in areas of ambiguity in data association around corners.

In summary, the posterior density is well represented by a relatively small term Gaussian mixture at each sample in the collected data. This compactness explains how the RT-GSF with 5 terms in the posterior density is able to outperform the PF with 2000 particles (Figure 2.6).

2.5.5 Artificial Increase in Likelihood of Particle Filter

An inexpensive, heuristic method for increasing the ability of a particle filter to represent the true posterior distribution is to artificially increase the size of the measurement likelihood, thereby changing the weights on the importance samples of the bootstrap PF. In this analysis, the component covariances (equation 2.5) are increased by a factor of 10 in the likelihood function. Figure 2.11 shows the histogram of differences between the RT-GSF (with the original measurement covariances) and the PF with an increased measurement likelihood uncertainty. On average, over the entire run, the RT-GSF performs $\bar{\Delta} = 0.28$ m better than the PF. When a paired t-test is performed, the 0.28 m average performance improvement of the RT-GSF is statistically significant at the $p = 0.05$ level. In fact, the t-test concludes the RT-GSF outperforms the PF with increased measurement variance over the 37 minute run by at least 0.27 m at the 5% significance level. The increased measurement uncertainty in the likelihood function allows the particle filter weights to avoid being overly peaked and misrepresenting the posterior distribution. Furthermore, this better representation allows the filter to avoid divergent behavior. This shows the PF to be more sensitive to the choice of likelihood model.

Another suggestion that the particle filter is not representing the true posterior distribution comes 120 seconds after the start of the GPS blackout when the vehicle approaches the second turn. Figure 2.12 shows an overhead view of the true pose and the estimated pose from the RT-GSF, the PF and the PF with increased measurement variance. Figure 2.12 shows the particle filter estimate diverges from the truth and, as a result, the vision measurements along the straight section of road are rejected as statistically improbable under the χ^2 hypothesis test from (2.33). This causes the filter to remain diverged along the entire straightaway. However,

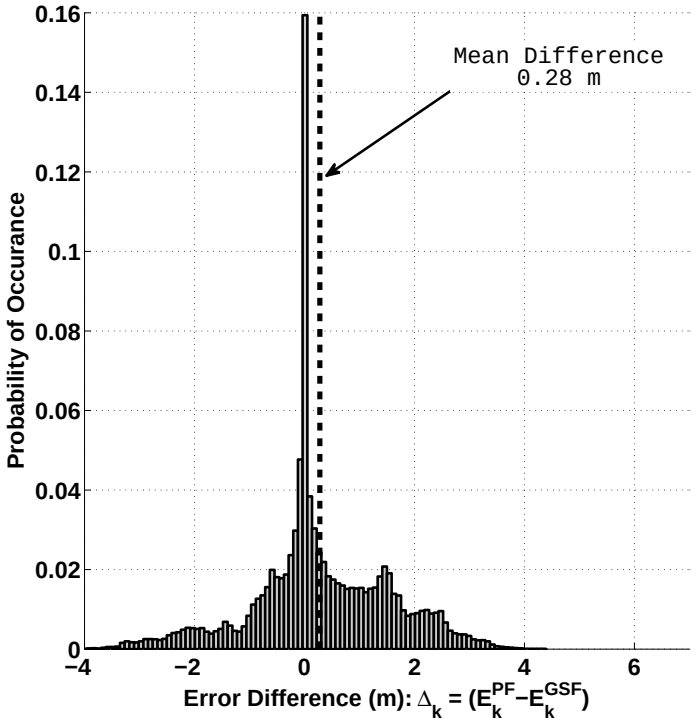

Figure 2.11: Histogram of the difference in the position error (Δ_k) between the RT-GSF and PF with a larger measurement variance. The PF has improved performance, but the RT-GSF still outperforms the PF with statistical significance.

the RT-GSF is able to incorporate the vision measurements and avoid diverging. The measurement rejection is a direct result of an incorrect representation of the posterior distribution, further showing the RT-GSF is more accurately representing the posterior distribution. It is noted that increasing the measurement likelihood by a factor of 10 in the posterior allows the particle filter to remain converged.

2.5.6 Increasing the Number of Particles in the PF

To evaluate the influence of increasing the number of particles on the size of the effective number of mixands in the Gaussian mixture approximation of the posterior,

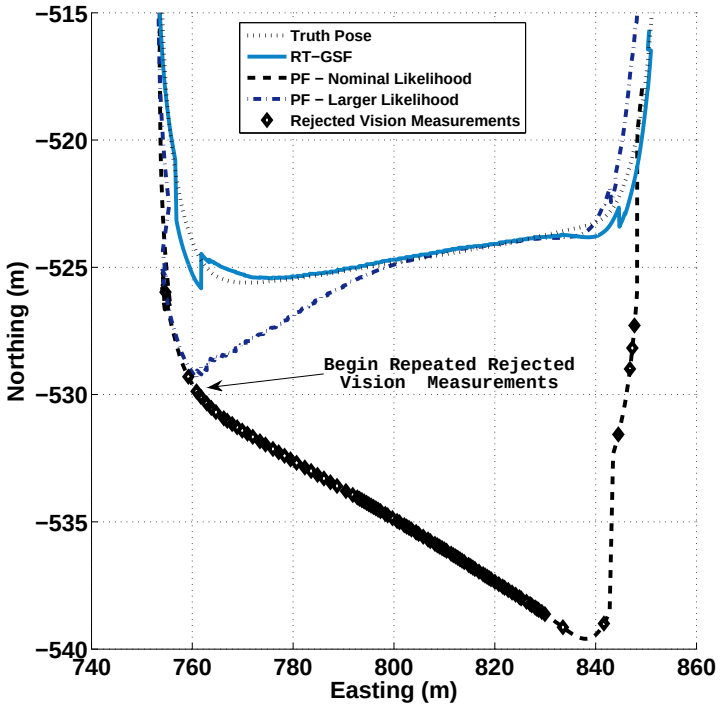

Figure 2.12: Plot of the RT-GSF and two variants of the PF, as the vehicle is turning through an intersection and onto a straight road. In the PF with the nominal likelihood, vision measurements are continuously rejected when the posterior distribution is not accurately representing the true error in the pose estimates.

the number of particles was varied over $1000 - 10000$ particles. Figure 2.13 shows the average and standard deviation of the difference ($\Delta_k = E_k^{\text{PF}} - E_k^{\text{GSF}}$) between the RT-GSF and the PF with varied number of particles . The plot shows the RT-GSF on average outperforms the PF. The paired t-test confirms the RT-GSF performance is statistically significant better than the PF at $p = 0.05$ confidence level. This demonstrates the 'solution' to increasing the performance of the PF is not to simply increase the number of particles. In fact, the RT-GSF is able to better represent the posterior distribution regardless of the number of particles used.

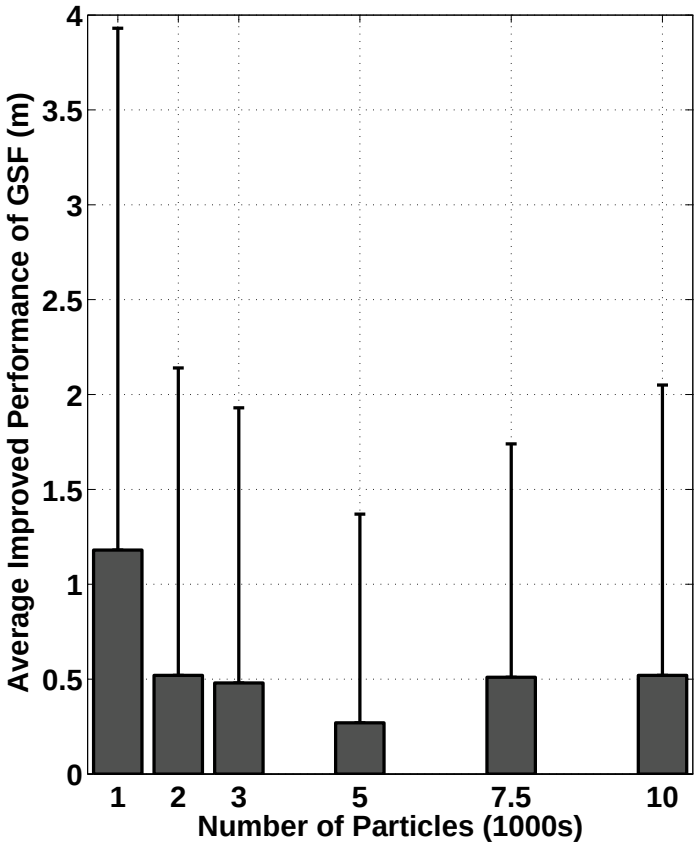

Figure 2.13: Plot of the average improved performance of the RT-GSF compared to the PF with increasing number of particles. The RT-GSF statistically outperforms the PF regardless of the number of particles used.

To further evaluate any consequences of the different number of particles, the percentage of measurements rejected as a result of the χ^2 hypothesis tests (2.33) are shown in Figure 2.14. The results highlight the impact of the posterior representation in each implementation. The RT-GSF has the lowest overall fraction of measurements rejected from the χ^2 hypothesis tests, which suggests the best representation of the true posterior density. The different in performance as a function of the different number of particles in the PF is also attributed to the fraction of measurements that is rejected. The implementations with few particles have

the most rejections. In addition, as the number of particles increases, the fraction of measurements rejected decreases, except for the implementation with $10,000$ particles. This shows a complex interaction between the hypothesis tests and the specific implementation. However, the hypothesis tests are necessary, because each implementation diverges without hypothesis testing. This is because there are numerous features in the world that are not represented in the known map, including unmapped stopped signs and railroad tracks that are often confused for stoplines or lane markers.

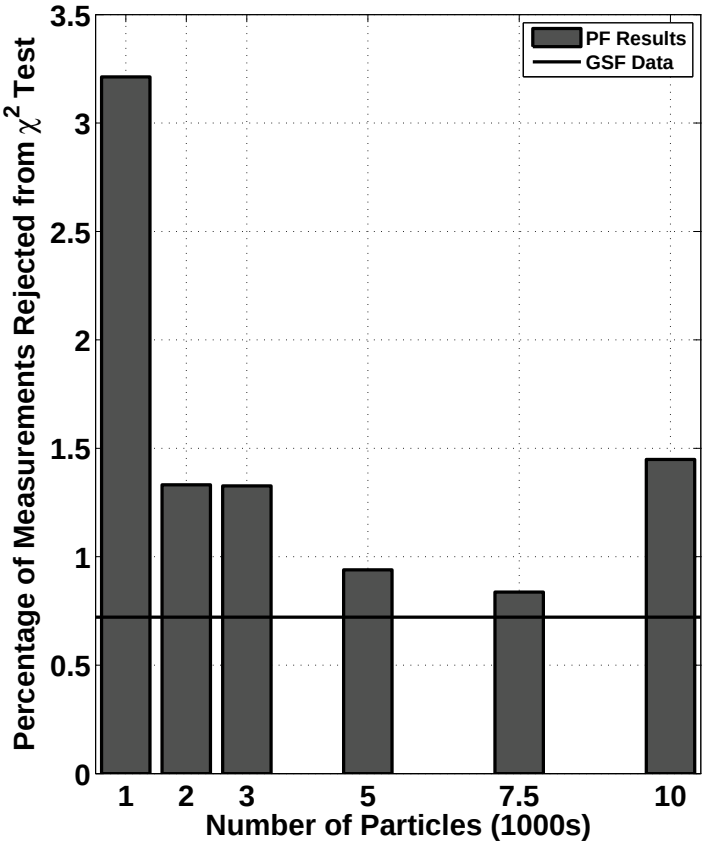

Figure 2.14: Plot of the percentage of measurements rejected under the χ^2 hypothesis test for different number of particles. The difference in performance from increasing the number of particles is attributed to the increased hypothesis rejections.

2.5.7 Environmental Sensitivities between RT-GSF and PF

To evaluate the posterior representation on different portions of the map, the number of Gaussians in the RT-GSF and effective number of Gaussians in the PF is examined based on the type of road the vehicle is traversing. Figure 2.15 shows the overall average number of Gaussian mixtures terms in the RT-GSF posterior prior to condensation for $N_{g_{\min}} = 2$ and $N_{g_{\max}} \in [2, 15]$ and PF posterior distribution for $1000 - 10000$ particles (horizontal axis). The different bars represent the different areas of the map, including straight roads, areas within 10 m of a corner and all areas of the map. Figure 2.15 shows the average number of Gaussians in the RT-GSF posterior prior to condensation increases linearly with the maximum number of allowed terms in the posterior after condensation. The number of terms in the RT-GSF prior to condensation is approximately the same on the straight-away road and in the corners. This is because the number of terms in the posterior density of the RT-GSF is allowed to adapt regardless of the location on the map. The increased average number of Gaussians in the posterior distribution as a function of $N_{g_{\max}}$ does not have an impact on performance (Figure 2.4), so a low number of terms is appropriate for this application.

However, for the PF effective Gaussian mixture, Figure 2.15 shows a larger difference in the straight-away section compared to the corners. This show when the vehicle was within 10 m of an intersection the complexity of the posterior distribution increases. This is expected behavior, because there is less uncertainty on the straight roads and more uncertainty about correct measurement-feature data assignment after exiting corners. Figure 2.15 shows that increasing the number of particles does not change the average effective number of mixands in the PF pos-

terior distribution. This shows the particle filter is not representing the posterior distribution effectively, because a small term Gaussian mixture is equivalent (in terms of performance) to the thousands of particles.

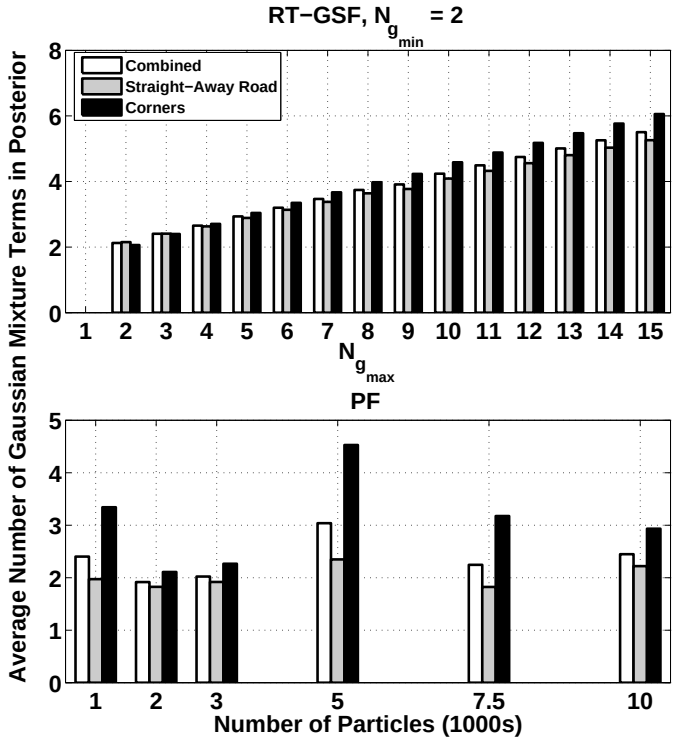

Figure 2.15: Plot of the average number of Gaussians in the posterior distribution of the RT-GSF and PF as a function of the vehicle location on the map. The average number of terms in the RT-GSF posterior increases linearly with $N_{g_{\max}}$ and is similar regardless of the location on the map. The average number of effective Gaussians in the PF data is small despite increasing the number of particles and has larger variation between the different sections of the map.

2.5.8 Computational Comparison Between RT-GSF and PF

To evaluate whether the enhanced performance of the RT-GSF is at the cost of additional computation, a timing analysis was conducted and compared against the bootstrap PF. The RT-GSF and PF are implemented in C++ and run on a dual-core PC. Figure 2.16 shows the computation time for the RT-GSF with different values of $N_{g_{\max}}$ for $N_{g_{\min}} = 5$ and the PF with different number of particles. The total computation time is shown along with the predict-update time and the resampling or condensation time. Note that the computation time is normalized by the total run-time of the particle filter with 2000 particles. The overall PF computation time increases quadratically in the number of particles. The overall RT-GSF computation time also increases quadratically, due to the condensation algorithm. However, the RT-GSF scales at a lower rate than the PF. This shows the added performance of the RT-GSF is not at the expense of increased computation.

2.6 Conclusion

A Real-Time implementation of a general Gaussian Sum filter was developed to perform localization in a known map using multi-modal vision measurements in the absence of GPS with map relative measurements. The RT-GSF uses Runnalls' condensation technique to maintain the number of Gaussian components in the posterior density over time; varying the number of terms according to the MMSE state covariance. The RT-GSF performance remains constant as a function of $N_{g_{\max}}$ as long as $N_{g_{\min}} > 1$. The RT-GSF remains converged with a precise global posi-

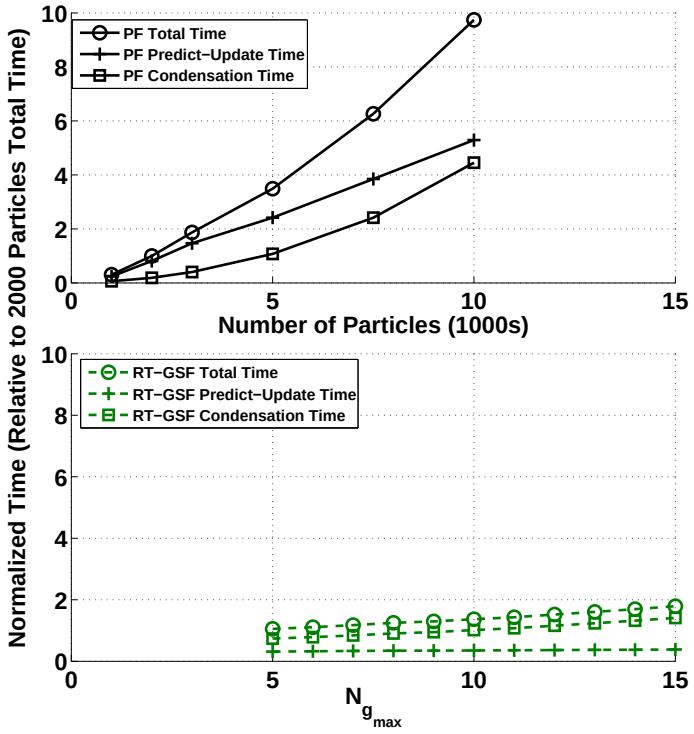

Figure 2.16: Plot of the RT-GSF and PF computational time as implemented in C++ and running on a dual-core PC. The computation scales quadratically for both filters, but the RT-GSF scales slower than the PF.

tion estimate over an extended 32 minute GPS blackout. The RT-GSF requires a multi-modal posterior distribution to avoid divergence, although a lower number of mixands in the posterior is required to solve the localization problem. The RT-GSF performs statistically better than the particle filter; in addition, using additional particles does not improve the filter performance due to complex interaction with the measurement hypothesis tests. The RT-GSF performance improvements are driven by time segments where the PF converges to the incorrect road lane. This is typically the result of a misrepresentation of the posterior distribution, which is further shown by the PF continuously rejecting vision measurements after divergence. Artificially increasing the likelihood variance in the PF enables the filter

to remain converged, but the RT-GSF is still able to perform statistically better. The PF posterior distribution is inefficient in that it can be sufficiently represented with a small number of terms in a Gaussian mixture, regardless of the number of particles used in the filter. The largest area of complexity in the PF posterior distribution is located around corners in the map, where few map-aided measurements are available. The RT-GSF and PF computation time both scale quadratically, but the RT-GSF scales at a slower rate. Finally, the added performance of the RT-GSF does not come at the expense of additional computation.

Acknowledgments

The authors would like to thank Peter Moran and Aaron Nathan for assistance in collecting data and the members of the Cornell DARPA Urban Challenge Team for providing sensors, sensor interfaces, and other data collection support. This work is supported under the Northrop Grumman Electronic Systems Scholars Program and ARO Grant #W911NF-09-1-0466.

CHAPTER 3

GENERALIZED DATA FUSION FOR DISTRIBUTED TERRAIN ESTIMATION

Abstract

A distributed grid-based terrain mapping algorithm using Gaussian Mixture Models is developed for use in tree connected and arbitrary connected sensor networks. The distributed data fusion rules are developed that operate directly on the sufficient statistics summarizing the grid-cell height and uncertainty. The exact solution, equivalent to a centralized solution, is achieved through use of channel filters if the sensor network topology is a tree. Three different tree topologies are investigated for the channel filter solution for evaluation of information gain. For arbitrary networks, a novel distributed data fusion solution is developed through use of a generalized data fusion technique based on combination of a sufficient statistic information set for each grid-cell. The computation of the generalized fusion weight is investigated and three techniques explored to compute the weight. For evaluating the distributed grid-based terrain mapping techniques two novel definitions of the cumulative information content in the map and the entire sensor network are developed. A bound on the maximum amount of information lost per grid-cell is derived for the generalized data fusion process. Finally, the maximum information loss metric is shown on the map and can be used to balance exploration and verification of areas with large uncertainty due to information loss. The distributed grid-based terrain mapping algorithm is demonstrated in an experimental environment involving 8 autonomous robots operating in an indoor environment for 120 seconds. The data are recorded and the algorithms demonstrated using tree connected and optimally connected network topologies and compared against

the exact centralized solution.

3.1 Introduction

Mobile robots require an accurate representation of their operating environment for path planning, localization, and exploration. The representation of the environment required by the robots depends on both the mission and capabilities of the mobile robot. Planetary exploration robots (Se et al., 2005), or Unmanned Ground Vehicles (UGVs) performing security functions (Ani Hsieh et al., 2007) require rich representations of the environment, including 3D terrain maps. The terrain maps can be generated via a variety of different sensors which provide depth information, including laser-line scanners, stereo vision, or depth from camera motion algorithms (Fregene et al., 2005).

Robots equipped with laser range finders that scan a single line have been used to generate 2D occupancy grid maps (Elfes, 1989) which represent the terrain as a planar grid, with each cell having a probability of being occupied or unoccupied. A natural extension of 2D occupancy grids was identified (Elfes, 1989) and implemented (Moravec, 1996) with stereo-vision using voxels (3D pixels) to represent the occupancy status of a given rectangular prism. Alternative techniques have been developed for terrain mapping that have a more compact representation than 3D occupancy grid maps. A technique was presented for translating point data, such as that obtained from laser line scanners, to triangular meshes that handle spurious data and holes in the point clouds (Roth and Wibowo, 1997). A 3D mapping technique was also introduced based on fine-grained multi-polygon surface models (Thrun et al., 2000). The polygon representation was made more compact (Thrun et al., 2004) by using a real-time variant of the EM algorithm to cluster range data and simultaneously estimate the number and shape of planes in the scan data. While these techniques provide rich and dense maps, they do not

provide online evaluation about the accuracy of the maps. The most compact true 3D occupancy grid work appears to be Octomap (Wurm et al., 2010) and provides rich 3D maps for large volumes with minimal memory requirements compared to the full 3D version; however, the limitation is that no accuracy is represented in the estimated map. Octomap does motivate a memory saving technique in the current grid-based implementation to use a quadtree that avoids instantiating cells until they are observed with the lidar sensor.

Other 3D mapping techniques include 2.5D representations that model the height of grid cells in a planar map assuming independence of heights in neighboring grid cells. A 2.5D map was created using laser range finders that assigns the maximum height return in each grid cell to the terrain height (Bares et al., 1989). A similar technique was used after creating Gaussian distributed terrain height estimates for each grid cell (Kleiner and Dornhege, 2007). The mixture-model based technique used here was introduced (Miller and Campbell, 2006), where the elevation of each grid cell is modeled as a mixture of Gaussian terrain height estimates and provides an estimate of the elevation uncertainty in each grid cell. A modified elevation map technique to handle environments with multiple dominant elevations was presented (Pfaff et al., 2007) where a dominant terrain height in each grid cell is selected. The proposed multi-level surface maps, where multiple elevations in a grid cell are represented by vertical and horizontal patches, was likewise presented (Triebel et al., 2006). A combination of the techniques of (Miller and Campbell, 2006) and (Triebel et al., 2006) was used to create probabilistic multi-level sets (Rivadeneyra and Campbell, 2011). An approach to incorporate pose uncertainty into Gaussian Processes used to represent terrain maps via contextual occupancy was introduced (O'Callaghan et al., 2010), but does not incorporate sufficient statistics and introduces a large computational burden. All of these mapping techniques

create dense maps that are different from those created from point obstacles (Lu and Milios, 1997) which are common in simultaneous localization and mapping (SLAM) procedures. Miller's mixture model based representation is used in this paper, because a dense elevation map is represented probabilistically; the development to multiple robots in arbitrary sensor networks is subsequently presented.

Multi-platform mapping has been performed using a variety of different techniques. In the context of SLAM, when the initial robot positions are known, or the robots are operating in a common coordinate system, multi-robot mapping is a simple extension of single robot mapping for centralized map creation (Fox et al., 2006). Multi-robot 3D mapping was performed where the 3D map (polygons) from each local sensor node is broadcast to all other nodes and the received local map is inserted into the global map and fused using a quadratic error measure (Thrun, 2001). This technique relies on each robot localizing themselves in the map of the other robot prior to transmitting map information, which may be made easier through localization with 2D bar codes (Olson, 2011). Also, 3D occupancy grid maps were joined from different sensor nodes through an exhaustive search process (Ryde and Hu, 2006). A technique based on spectral information in occupancy grid maps was used to fuse local maps into a global estimate (Carpin, 2008). Another technique forced each robot to share a common map while performing marginal-SLAM, and fuses maps by matching features (Martinez-Cantin et al., 2007). Another multi-robot SLAM technique does not require the initial positions of the robots to be known, but instead have the robots actively seek each other to determine their relative locations in order to fuse maps (Fox et al., 2006). All of these approaches essentially insert local maps into a global map, but none allow uncertainty or confidence associated with the map to be updated in the process and none ensure consistency in the presence of loops in the local sensor

network without data-tagging (Ceruti et al., 2006).

The objective of sensor networks employing distributed data fusion (DDF) networks is to generate globally consistent estimates on each sensor node without the use of a centralized processing node, without a common communication bus (node-to-node communication only) and without requiring each node to have global knowledge of the network topology (neighborhood knowledge only) (Nettleton et al., 2000). The advantages of DDF in a sensor network include robustness and modularity (Makarenko and Durrant-Whyte, 2006). The channel filter (Grime et al., 1992) (Grime and Durrant-Whyte, 1994) is a convenient approach to performing DDF for general probability distribution representations (Makarenko and Durrant-Whyte, 2006). None of the dense multi-robot mapping techniques described above adhere to the DDF paradigm. However, discrete feature maps, such as those generated in the SLAM paradigm, can be fused in a distributed data fusion paradigm (Nettleton et al., 2000).

In an distributed data fusion architecture that does not propagate measurements, the optimal fusion solution must factor common information (Liggins et al., 1997). An earlier approach based on linear opinion pools was used to merge with a weighted average two distributions (Genest and Zidek, 1986) The channel filter approach accomplishes this only when the global network topology is acyclic or tree connected (Grime and Durrant-Whyte, 1994). For maximally robust sensor networks (Dekker and Colbert, 2004), cyclic topologies are preferred. In order to avoid data-tagging or sending pedigree information along with each estimate, generalized data fusion techniques with unknown correlations are required. An early approach for conservative fusion of Gaussian distributions with unknown correlation was the covariance intersection approach (Julier and Uhlmann, 1997).

The covariance intersection method was extended to general distributions (Mahler, 2000) and recognized as similar to the Chernoff information metric (Hurley, 2002) that finds a fused distribution half-way (in the Kullback-Leibler sense) between the two distributions to fuse (Cover and Thomas, 1991). The Chernoff fusion approach was studied for use with Gaussian Mixtures and several approximations introduced (Julier, 2006), the most accurate was a pseudo-Chernoff covariance intersection technique. The generalized Chernoff fusion technique for fusing multiple distributions with unknown correlations without performing sequential Chernoff fusion was subsequently introduced (Farrell and Ganesh, 2009). Additional recent techniques include an effort to estimate and exploit common information (Julier, 2009). A fusion approach using Schur dominance was introduced for discrete distributions, and is more accurate than Chernoff fusion because it can find solutions off the Kullback-Leibler chord between the two distributions (Rendas and Leita, 2010). A message passing approach that solves the double counting problem in 2-tree network topologies was presented (Thompson and Durrant-Whyte, 2010). The generalized data fusion approach here is a hybrid approach between linear opinion pools and generalized Chernoff fusion, because it operates on a set of sufficient statistics, but also maintains a conservative approximation, without double counting mutual information.

This paper develops an algorithm for generating dense terrain maps from multiple robots in a common coordinate system in arbitrary networks. It expands an earlier conference paper (Schoenberg and Campbell, 2009) that considers distributed fusion in only tree connected networks with a limited data set. The terrain maps local to each sensor are generated using Miller's mixture-model based approach, which is good for memory scaling, accuracy, and formally dealing with measurement errors introduced by the robot pose estimation and sensor.

Two distributed grid-based terrain mapping algorithms are developed for use in multi-robot mapping. Compact information sets representing the in-cell height estimate are fused in a distributed sensor network. The first algorithm uses a channel filter for DDF and three different tree topologies (chain, star and bridge) are utilized and the impact on information gain based on topology is evaluated with insights into information gain rates based on topology. The second distributed mapping technique is a novel approach for consistent, but conservative fusion for arbitrary network topologies based on fusion of sufficient statistics. The generalized distributed terrain mapping technique is evaluated on an optimally connected network. A bound on the information loss is derived and shown to be a function of the terrain map location; this representation will be potentially important in planning. For all distributed fusion mapping approaches, each local sensor node updates and maintains a terrain map and formally incorporates accuracy information from other sensor nodes in a distributed fashion. The algorithms are examined using a laboratory experiment and the generation of quality terrain maps with consistent, but conservative fusion is observed.

The remainder of the paper includes a summary of the mixture-model based terrain estimation algorithm for a single sensor node in Section 3.2. Section 3.3 extends the algorithm to the distributed data fusion paradigm for tree connected topologies using the channel filter in Section 3.3.1 and arbitrary topologies using a generalized fusion rule in Section 3.3.2 along with the maximum information loss bound. Sections 3.4 and 3.5 describe the laboratory experiments and results, and Section 3.6 finishes with conclusions.

3.2 Grid-Based Terrain Mapping Using Gaussian Mixture Models

The mixture-model based terrain estimation algorithm introduced by Miller and Campbell (Miller and Campbell, 2006) translates laser scanner terrain detections into an elevation distribution for the height of the terrain in each cell in a planar grid. The algorithm begins by transforming the terrain detections into an inertial coordinate system accounting for uncertainties in the sensor alignments and measurements errors. Next, the terrain detections are probabilistically associated to cells in the terrain grid. An estimate of the elevation distribution in the cell is then generated from the measurements associated in each grid cell. The elevation density in each grid cell is assumed independent from one grid cell to the next. The key feature of (Miller and Campbell, 2006) is representing each cell as a mixture, which greatly eases the ability to accumulate additional measurements into the terrain estimate, as shown in the remainder of this section.

The mixture model terrain estimation algorithm in (Miller and Campbell, 2006) begins by defining a planar grid of N_c cells in an inertial coordinate system. Each rectangular grid cell is defined such that the l^{th} cell extends from E_{l-} to E_l+ in the Easting direction and N_{l-} to N_{l+} in the Northing direction. This places the center of grid cell l at $EN_l = [E_l \; N_l]^T$ where $E_l = \frac{1}{2}(E_{l-} + E_{l+})$ and $N_l = \frac{1}{2}(N_{l-} + N_{l+})$. The extent of each grid cell is also easily defined such that $\Delta E_l = (E_{l+} - E_{l-})$ and $\Delta N_l = (N_{l+} - N_{l-})$. The planar grid need not have a uniform cell size, but it is convenient to assume one without loss of generality.

The transformation of range and angle laser scanner measurements $\underline{r}$ to an inertial Cartesian coordinate system requires knowing the orientation of the sensor,

defined by parameters $\underline{p}$, in the inertial coordinate system. For an East-North-Up (ENU) coordinate system, the transformed laser measurements are defined by a nonlinear function $f(\underline{p}, \underline{r})$ defined in (3.1).

$$\underline{r}^{ENU} \triangleq \begin{bmatrix} E \\ N \\ U \\ 1 \end{bmatrix} = f(\underline{p}, \underline{r}) \tag{3.1}$$

where defining $\underline{r}^{ENU}$ as a 4×1 vector allows for the use of 4×4 transformation matrices when converting between coordinate systems.

Unfortunately, no sensor measurement or alignment parameters are perfect. In fact, errors can arise from a variety of reasons including miss-calibration, thermal noise, and encoder quantization for gimbaled sensors. The observed sensor orientation parameters and laser measurements are corrupted by noise from the truth as defined in (3.2) and (3.3).

$$\hat{\underline{p}} = \underline{p} + \nu_{\underline{p}} \tag{3.2}$$

$$\hat{\underline{r}} = \underline{r} + \nu_{\underline{r}} \tag{3.3}$$

where $\nu_{\underline{p}}$ and $\nu_{\underline{r}}$ are the Gaussian distributed measurement errors that are assumed independent from each other with zero mean and covariances $Q_{\underline{p}}$ and $Q_{\underline{r}}$.

The transformed laser measurements with errors, are defined by $f(\underline{p}+\nu_{\underline{p}}, \underline{r}+\nu_{\underline{r}})$, which is linearized about the observed values $\hat{\underline{p}}$ and $\hat{\underline{r}}$ in (3.2) and (3.3):

$$\underline{r}^{ENU} \approx f(\hat{\underline{p}}, \hat{\underline{r}}) + \left.\frac{\partial f}{\partial \underline{p}}\right|_{\underline{p}=\hat{\underline{p}}, \underline{r}=\hat{\underline{r}}} \nu_{\underline{p}} + \left.\frac{\partial f}{\partial \underline{r}}\right|_{\underline{p}=\hat{\underline{p}}, \underline{r}=\hat{\underline{r}}} \nu_{\underline{r}} \tag{3.4}$$

$$= f(\hat{\underline{p}}, \hat{\underline{r}}) + J_{\underline{p}}(\hat{\underline{p}}, \hat{\underline{r}})\nu_{\underline{p}} + J_{\underline{r}}(\hat{\underline{p}}, \hat{\underline{r}})\nu_{\underline{r}} \tag{3.5}$$

where $J_{\underline{p}}$ and $J_{\underline{r}}$ are the Jacobians of the measurement function f taken with respect to $\underline{p}$ and $\underline{r}$.

Using the linearized measurement function, it is possible to generate the posterior distribution of the terrain detection in the inertial coordinate system $r^{ENU} \sim p(E,N,U)$ that is approximated with a Gaussian distribution $p(E,N,U) \approx \mathcal{N}(\hat{r}^{ENU}, P_{\hat{r}}^{ENU})$. The mean and covariance of the posterior distribution of the terrain detection is given in (3.6) and (3.7), conditioned on all the information available (I).

$$\hat{r}^{ENU} = \begin{bmatrix} \hat{e} \\ \hat{n} \\ \hat{u} \\ 1 \end{bmatrix} = E\left[\underline{r}^{ENU}|I\right] = f(\underline{\hat{p}}, \underline{\hat{r}}) = \begin{bmatrix} \hat{r}^{EN} \\ \\ \hat{u} \end{bmatrix} \tag{3.6}$$

$$\begin{aligned} P_{\hat{r}}^{ENU} &= E\left[(\underline{r}^{ENU} - \underline{\hat{r}}^{ENU})(\underline{r}^{ENU} - \underline{\hat{r}}^{ENU})^T|I\right] \\ &= J_{\underline{p}}(\underline{\hat{p}}, \underline{\hat{r}}) Q_{\underline{p}} J_{\underline{p}}^T(\underline{\hat{p}}, \underline{\hat{r}}) + J_{\underline{r}}(\underline{\hat{p}}, \underline{\hat{r}}) Q_{\underline{r}} J_{\underline{r}}^T(\underline{\hat{p}}, \underline{\hat{r}}) \\ &= \begin{bmatrix} P_{\hat{r}}^{EN} & P_{\hat{r}}^{EN,u} \\ P_{\hat{r}}^{u,EN} & P_{\hat{r}}^{u} \end{bmatrix} \end{aligned} \tag{3.7}$$

where $\hat{r}^{ENU}$ and $P_{\hat{r}}^{ENU}$ are written in block form to identify specific portions that are used later in the algorithm. Equations (3.6) and (3.7) represent the terrain detection with Gaussian uncertainty associated with the source; the next step is to perform measurement-to-grid association.

The U dimension is marginalized out of each one of the M transformed terrain measurements $\{\hat{r}_m^{ENU}, P_{\hat{r}_m}^{ENU}\}_{m=1}^{M}$ in order to generate a planar distribution of the terrain measurement uncertainty (3.8):

$$\begin{aligned} p_m(E,N) &= \int_{-\infty}^{\infty} p_m(\underline{r}) dU = \int_{-\infty}^{\infty} p_m(E,N,U) dU \\ &= \int_{-\infty}^{\infty} \mathcal{N}\left(\hat{r}_m^{ENU}, P_{\hat{r}_m}^{ENU}\right) dU \\ &= \mathcal{N}\left(\hat{r}_m^{EN}, P_{\hat{r}_m}^{EN}\right) \end{aligned} \tag{3.8}$$

where each of the M terrain measurements from a single scan of the laser range finder is assumed independent.

This gives the in-plane Gaussian distribution of the terrain measurement, where $\hat{r}_m^{EN}$ and $P_{\hat{r}_m}^{EN}$ are the mean and covariance respectively. The probability of the terrain measurement originating from a specific cell l is then computed as:

$$\begin{aligned} \Pr\left(\hat{r}_m^{EN} \in l\right) &= \int_{E_{l-}}^{E_{l+}} \int_{N_{l-}}^{N_{l+}} p_m(E,N) dN dE \\ &\approx \left(\Delta E_l\right)\left(\Delta N_l\right) p_m(E_l, N_l) \end{aligned} \tag{3.9}$$

where the probability is approximated using a single Riemann sum.

Next, the in-cell height distribution of the m^{th} terrain measurement given it occurred at the center of the l^{th} grid cell is computed. The univariate in-cell height distribution is found by taking the conditional distribution of the Gaussian defined by the mean (3.6) and covariance (3.7):

$$p(u_{m\in l} | E = E_l, N = N_l) \approx \mathcal{N}\left(\hat{u}_{m\in l}, \sigma^2_{\hat{u}_{m\in l}}\right) \tag{3.10}$$

where the mean and covariance are found via standard conditioning rules of the Gaussian distribution (Bar-Shalom et al., 2001):

$$\hat{u}_{m\in l} = \hat{u}_m + P_m^{u,EN}\left(P_m^{EN}\right)^{-1}\left[EN_l - \hat{r}_m^{EN}\right] \tag{3.11}$$

$$\sigma^2_{\hat{u}_{m\in l}} = P_m^u - P_m^{u,EN}\left(P_m^{EN}\right)^{-1} P_m^{EN,u} \tag{3.12}$$

Now, the set of all laser scanner measurements $\underline{r}^K \triangleq \left\{\left\{\rho_{km}, \theta_{km}\right\}_{m=1}^M\right\}_{k=0}^K$ up to time K is defined, assuming M measurements per scan of the range ρ_{km} and angle θ_{km}, along with a set of sensor alignment measurements $\underline{p}^K = \left\{\underline{p}_k\right\}_{k=0}^K$. Each laser scanner measurement in the set $\underline{r}^K$ and sensor alignment measurement $\underline{p}^K$ have been mapped to each grid cell l with an association probability and height

estimate $\left\{p_{km\in l}, \hat{u}_{km\in l}, \sigma^2_{\hat{u}_{km\in l}}\right\}_{k=0}^{K}$. In practice, it is only necessary to compute the association probability and height estimates for a certain number of cells around the original laser scanner measurement(Miller and Campbell, 2006).

The next step is to compute the distribution of the elevation in the lth cell given all of the terrain detections, denoted as $p(U_l|\underline{r}^K, \underline{p}^K)$. This is approximated as a Gaussian mixture:

$$p(U_l|\underline{r}^K, \underline{p}^K) \approx \frac{1}{c_l}\sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}\cdot\mathcal{N}\left(\hat{u}_{km\in l}, \sigma^2_{\hat{u}_{km\in l}}\right) \tag{3.13}$$

where $c_l = \sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}$ is a normalizing constant. Considering just the first and second moments of this Gaussian mixture, the lth cell's terrain height is categorized by the mean and covariance of the Gaussian mixture (Bar-Shalom et al., 2001):

$$\hat{U}_{GM_l} = \frac{1}{c_l}\sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}\hat{u}_{km\in l} \tag{3.14}$$

$$\sigma^2_{\hat{U}_{GM_l}} = \frac{1}{c_l}\sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}\left(\hat{u}^2_{km\in l} + \sigma^2_{\hat{u}_{km\in l}}\right) - \hat{U}^2_{GM_l} \tag{3.15}$$

This allows each sensor node to maintain a recursively calculated information set with sufficient statistics for each grid cell, as:

$$\begin{aligned} Z_l^K \triangleq \Bigg\{ & Z_p^K = \sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l},\ Z_{p\hat{u}}^K = \sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}\hat{u}_{km\in l}, \\ & Z_{p\hat{u}^2}^K = \sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}\hat{u}^2_{km\in l},\ Z_{p\sigma^2_{\hat{u}}}^K = \sum_{k=0}^{K}\sum_{m=1}^{M} p_{km\in l}\sigma^2_{\hat{u}_{km\in l}} \Bigg\} \end{aligned} \tag{3.16}$$

From these sufficient statistics, the Gaussian mixture first and second moment are computed:

$$\hat{U}_{GM_l} = \frac{1}{Z_p^K}\cdot Z_{p\hat{u}}^K \tag{3.17}$$

$$\sigma^2_{\hat{U}_{GM_l}} = \frac{1}{Z_p^K}\cdot\left(Z_{p\hat{u}^2}^K + Z_{p\sigma^2_{\hat{u}}}^K\right) - \left(\frac{Z_{p\hat{u}}^K}{Z_p^K}\right)^2 \tag{3.18}$$

The information set Z_l^K is defined to be all of the accumulated information from the initial time up to and including time t_k. The update of the information set takes a simple recursive form for subsequent time steps; for example, the update for κ additional time steps of data is:

$$\begin{aligned} Z_p^\kappa &= Z_p^K + Z_p^{K+1,K+\kappa} \\ &= Z_p^K + \sum_{k=K+1}^{K+\kappa} \sum_{m=1}^{M} p_{km\in l} \end{aligned} \tag{3.19}$$

$$\begin{aligned} Z_{p\hat{u}}^\kappa &= Z_{p\hat{u}}^K + Z_{p\hat{u}}^{K+1,K+\kappa} \\ &= Z_{p\hat{u}}^K + \sum_{k=K+1}^{K+\kappa} \sum_{m=1}^{M} p_{km\in l}\hat{u}_{km\in l} \end{aligned} \tag{3.20}$$

$$\begin{aligned} Z_{p\hat{u}^2}^\kappa &= Z_{p\hat{u}^2}^K + Z_{p\hat{u}^2}^{K+1,K+\kappa} \\ &= Z_{p\hat{u}^2}^K + \sum_{k=K+1}^{K+\kappa} \sum_{m=1}^{M} p_{km\in l}\hat{u}_{km\in l}^2 \end{aligned} \tag{3.21}$$

$$\begin{aligned} Z_{p\sigma_{\hat{u}}^2}^\kappa &= Z_{p\sigma_{\hat{u}}^2}^K + Z_{p\sigma_{\hat{u}}^2}^{K+1,K+\kappa} \\ &= Z_{p\sigma_{\hat{u}}^2} + \sum_{k=K+1}^{K+\kappa} \sum_{m=1}^{M} p_{km\in l}\sigma_{\hat{u}_{km\in l}}^2 \end{aligned} \tag{3.22}$$

The information set for each grid cell allows a dense map with probabilistic accuracy information to be stored using four constants for each grid cell. More importantly, as new independent information becomes available from the local sensor node or remote sensor nodes, the information sets are updated with (3.19)-(3.22) and constant memory for a given grid cell is used.

In addition, an information set for the centralized solution across the entire sensor network is computed by summing the local node information sets (3.16) from each of the N nodes together:

$$\mathcal{Z} = \sum_{n=1}^{N} Z_n \tag{3.23}$$

3.3 Distributed Terrain Mapping Using Gaussian Mixture Models

The distributed terrain estimation algorithm proposed here allows each sensor node to maintain an estimate of the global environment, in a form identical to the local mixture based terrain map on a single sensor node. The distributed data fusion approaches eliminate a central fusion center, to provide enhanced scalability and survivability in the presence of network disturbance. In addition, distributed data fusion requires only local knowledge of the network topology at a given node, instead of requiring global knowledge. Two approaches are developed here for distributed terrain estimation. The first maintains an optimal distributed solution that is equivalent to the centralized solution, but requires a tree network topology. The second approach is a distributed solution that is statistically consistent, avoids rumor propagation and is used for arbitrary network topologies, but is conservative in its estimate.

3.3.1 Optimal Distributed Solution Using the Channel Filter

A goal for distributed data fusion is that as all information is communicated around the network, the distributed solution approaches the centralized data fusion solution. The channel filter (Makarenko and Durrant-Whyte, 2006) provides a convenient framework to perform DDF. The channel filter keeps track of common information shared between sensor nodes on opposite ends of a data link. The network topology influences the implementation of the channel filter, and in the

proposed approach, a tree-connected, acyclic network topology is assumed. The desire is for each sensor node i and j on a communication link to estimate the state x (terrain height of an individual cell) given the union of the information sets available at each node (Makarenko and Durrant-Whyte, 2006):

$$p(x|Z_i^K \bigcup Z_j^K) = \frac{1}{c} \frac{p(x|Z_i^K)p(x|Z_j^K)}{p(x|Z_i^K \bigcap Z_j^K)} \tag{3.24}$$

where $p(x|Z_i^K)$ and $p(x|Z_j^K)$ are the posterior distributions including all information received at sensor nodes i and j, and $p(x|Z_i^K \bigcap Z_j^K)$ is the posterior distribution given all the common information contained in both information sets. One description of the channel filter (Makarenko and Durrant-Whyte, 2006) assumes the posterior distributions are Gaussian and uses the information form of the Kalman filter to maintain an estimate of (3.24). A similar technique could be used here for each grid cell, where the mean (3.14) and covariance (3.15) of the Gaussian mixture are used in the channel filter that assumes Gaussian posterior distribution. Fortunately, the mixture-model algorithm represents the terrain map with a succinct information set in each grid cell that enables tracking the union of the information sets in sensor nodes i and j directly using the channel filter.

The union of the information on a communication channel between sensor node i and j up to time k, is given by (Makarenko and Durrant-Whyte, 2006):

$$Z_i^K \bigcup Z_j^K = Z_{i\setminus j}^K + Z_{j\setminus i}^K + Z_{i\bigcap j}^K \tag{3.25}$$

$$= Z_i^K + Z_j^K - Z_{i\bigcap j}^K \tag{3.26}$$

where $Z_{i\bigcap j}^K$ is the common information contained in sets Z_i^K and Z_j^K. The tree-connected network topology is now advantageous, because all common information between nodes i and j is assured to have come across the $i-j^{\text{th}}$ communication link. Therefore, the common information up to time K is the union of the information

shared previously:

$$Z_{i\bigcap j}^{K} = Z_{i}^{K-1} \bigcup Z_{j}^{K-1} \tag{3.27}$$

The local information set at node i up to time k, given all of the information sets in the neighborhood N_i (sensor nodes connected to i) where $i \notin N_i$ can now be updated:

$$Z_{i}^{K} = Z_{i}^{K-1} + Z_{i}^{K-1,K} + \sum_{j \in N_i} \left[\tilde{Z}_{i\leftarrow j}^{k} - Z_{i\bigcap j}^{K-1} \right] \tag{3.28}$$

where $Z_{i}^{K-1,K}$ is the new information accumulated at sensor i locally (i.e. from new laser measurements) from t_{k-1} to t_k and $\tilde{Z}_{i\leftarrow j}^{k}$ is the information received at node i from neighboring sensor node j at time t_k.

The channel filter is robust to communication failure and does not need to verify receipt of a transmitted message from one node to another. Instead, the channel filter operates consistently if two rules are followed. First, each sensor node vows not to send back any information already received over a given channel:

$$\tilde{Z}_{i\rightarrow j}^{k} = Z_{i}^{K} - Z_{i\bigcap j}^{K-1} \tag{3.29}$$

$$= Z_{i}^{K} - Z_{i\leftarrow j}^{K-1} \tag{3.30}$$

where, $Z_{i\leftarrow j}^{K-1}$ indicates the information received at node i from node j up to t_{k-1}, this may not be the same as $Z_{j\rightarrow i}^{K-1}$, because of communication failure. Second, each sensor node is required to keep track of all information received from a given node. These two rules allow the estimate of the common information between nodes i and j at node i to be a simple assignment of the received information:

$$Z_{i\bigcap j}^{K} = \begin{cases} \tilde{Z}_{i\leftarrow j}^{k} & \text{if} \quad \tilde{Z}_{i\leftarrow j}^{k} \neq \emptyset \\ Z_{i\bigcap j}^{K-1} & \text{if} \quad \tilde{Z}_{i\leftarrow j}^{k} = \emptyset \end{cases} \tag{3.31}$$

These rules ensure that the channel filter always accounts for the common information and the unique information contained at each node. The consequence is that at each node, the estimate of the common information are not necessarily symmetric across the $i - j^{th}$ channel depending on communication loss:

$$Z_{i\bigcap j}^{K} \neq Z_{j\bigcap i}^{K} \tag{3.32}$$

where the first node index indicates the estimation of common information between nodes i and j resides on that particular node.

In the case of communication failure, the local information set at node i is updated (3.28) without information $\tilde{Z}_j^k$ from nodes whose communication links have failed such that the updated information is $Z_i^K = Z_i^{K-1} + Z_i^{K-1,K}$. The channel filter for the $i - j^{\text{th}}$ communication link remains unchanged while the link is down according to (3.31). If communication failure persists for κ time steps, the local sensor nodes continue to update local information (3.28) and when communication is restored, the information from neighboring nodes $\tilde{Z}_j^{K+\kappa}$ is received and assimilated at node i into $Z_i^{K+\kappa}$.

Theorem 3.3.1. *The channel filter counts independent and mutual information exactly once regardless of communication failure without message receipt notifications.*

Proof. Decompose the information contained at node i into constituent parts with the last successful communication from i to j n time steps ago, and m time steps from j to i.

$$Z_i^K = Z_{i\backslash j}^K + Z_{i\leftarrow j}^{K-m} + Z_{j\leftarrow i}^{K-n} \tag{3.33}$$

$$\tag{3.34}$$

Node i communicates with node j, following rule one (3.29) does not send back any previously received information:

$$\tilde{Z}^k_{i\to j} = Z^K_i - Z^{K-m}_{i\leftarrow j} \tag{3.35}$$

$$= Z^K_{i\setminus j} + Z^{K-n}_{j\leftarrow i} \tag{3.36}$$

Node j assimilates the received information from node i but subtracting any previously received information:

$$Z^K_j = Z^{K-1}_j + Z^{K-1,K}_j + \left[\tilde{Z}^k_{j\to i} - Z^{K-n}_{j\bigcap i}\right] \tag{3.37}$$

$$= Z^K_{j\setminus i} + Z^K_{i\setminus j} + Z^{K-n}_{j\leftarrow i} + Z^{K-m}_{i\leftarrow j} \tag{3.38}$$

$$= Z^K_{j\setminus i} + Z^K_{j\leftarrow i} + Z^{K-m}_{i\leftarrow j} \tag{3.39}$$

□

Theorem 3.3.1 shows that by following these two rules (3.29) and (3.31), the channel filter maintains a consistent estimate without double counting any information at all nodes regardless of communication failure and without message arrival confirmation notifications.

To summarize the distributed terrain algorithm, the information set for each grid cell is updated and received information assimilated locally (3.28), while the channel filter keeps track of the common information (3.31) on any communication link for each of the four parameters in each grid cell. Using the locally updated and assimilated information set (3.28), the final distribution of the in-cell terrain height estimate is computed using (3.14) and (3.15). The channel filter implementation used here requires a tree-connected topology to ensure all information received from node j at node i comes across link $i-j$. In addition, sensor node i must maintain a channel filter for each neighboring node. While the global network topology is

unknown to each node, it must be specified to maintain a tree topology to avoid double counting information, even with the channel filter. The tree topology is brittle, because any node or communication failure creates a disconnected network. To avoid these short-coming and increase robustness and survivability, a general data fusion solution is developed for the distributed terrain mapping.

3.3.2 Generalized Distributed Solution

The general distributed data fusion solution is designed to operate in arbitrary network topologies without need for knowledge of the global network topology. In addition, the distributed system desires to avoid data-tagging (Ceruti et al., 2006) and only transmit information after assimilation into the local sensor node. The general solution should also maintain consistency and avoid double counting information in the fused solution.

A number of possible generalized data fusion rules are possible, including the weighted exponential product from (Mahler, 2000):

$$p(x|Z_A^K \bigcup Z_B^K) = \frac{1}{c} p(x|Z_A^K)^{\omega} \cdot p(x|Z_B^K)^{(1-\omega)} \tag{3.40}$$
$$0 \leq \omega \leq 1$$

Another option is linear opinion pools created from a linear combination of probability densities (Genest and Zidek, 1986):

$$p(x|Z_A^K \bigcup Z_B^K) = \omega \cdot p(x|Z_A^K) + (1-\omega) \cdot p(x|Z_B^K) \tag{3.41}$$
$$0 \leq \omega \leq 1$$

which for each grid-cell j, is a linear combination of the Gaussian mixture (3.13) from sensor nodes A and B.

Unfortunately, the fusion approaches in (3.40) and (3.41) lead to intractable computations and an inability to maintain an additive sufficient statistic set (3.16) for the fused density. Instead, an alternative is to operate directly on each of the four sufficient statistics (3.16):

$$Z_f = \omega Z_i + (1-\omega) Z_j \tag{3.42}$$
$$0 \leq \omega \leq 1$$

which provides a means to fuse the sufficient statistics directly, instead of the underlying probability distributions in each grid cell in a consistent manner.

Theorem 3.3.2. *The linear combination fusion rule* (3.42) *discounts exclusive information, but conserves all common information in the sufficient statistics for the first and second moments of the fused Gaussian mixture.*

Proof.

$$Z_i \triangleq Z_{i\backslash j} + Z_{i \bigcap j} \tag{3.43}$$
$$Z_j \triangleq Z_{j\backslash i} + Z_{i \bigcap j} \tag{3.44}$$
$$Z_f = \omega Z_i + (1-\omega) Z_j \tag{3.45}$$
$$= \omega Z_{i\backslash j} + (1-\omega) Z_{j\backslash i} + Z_{i \bigcap j} \tag{3.46}$$

□

After applying these fusion rules between nodes i and j, the Gaussian mixture mean and covariance for the l^{th} cell at time t_k are computed using the same

sufficient statistics (3.16):

$$\hat{U}_{GM_{f_l}} = \frac{1}{Z^K_{p_{f_l}}} \cdot Z^K_{p\hat{u}_{f_l}} \tag{3.47}$$

$$= \frac{1}{\omega Z^K_{p_{i_l}} + (1-\omega) Z^K_{p_{j_l}}} \cdot \left(\omega Z^K_{p\hat{u}_{i_l}} + (1-\omega) Z^K_{p\hat{u}_{j_l}} \right) \tag{3.48}$$

$$\tag{3.49}$$

$$\sigma^2_{\hat{U}_{GM_{f_l}}} = \frac{1}{Z^K_{p_{f_l}}} \cdot \left(Z^K_{p\hat{u}^2_{f_l}} + Z^K_{p\sigma^2_{\hat{u}_{f_l}}} \right) - \hat{U}^2_{GM_{f_l}} \tag{3.50}$$

$$= \frac{\omega \left(Z^K_{p\hat{u}^2_{i_l}} + Z^K_{p\sigma^2_{\hat{u}_{i_l}}} \right) + (1-\omega) \left(Z^K_{p\hat{u}^2_{j_l}} + Z^K_{p\sigma^2_{\hat{u}_{j_l}}} \right)}{\omega Z^K_{p_{i_l}} + (1-\omega) Z^K_{p_{j_l}}} - \dots$$

$$\left(\frac{\omega Z^K_{p\hat{u}_{i_l}} + (1-\omega) Z^K_{p\hat{u}_{j_l}}}{\omega Z^K_{p_{i_l}} + (1-\omega) Z^K_{p_{j_l}}} \right)^2 \tag{3.51}$$

This results in a unique fusion result by operating directly on the sufficient statistics and is not equivalent to performing data fusion according to (3.40) nor (3.41) nor covariance intersection on the corresponding Gaussian mean and covariance for each cell on sensor nodes i and j. There are multiple ways to select ω in the fusion rule for the sufficient statistics. Several possibilities include a fixed ω (3.52), a weighted average of the fusion nodes grid cell association probabilities Z_p (3.53) or a minimization of the fused covariance $\sigma^2_{\hat{U}_{GM_f}}$ from (3.51) for each cell (3.54):

$$\omega_l = \text{constant} \tag{3.52}$$

$$\omega_l = \frac{Z^K_{p_{i_l}}}{Z^K_{p_{i_l}} + Z^K_{p_{j_l}}} \tag{3.53}$$

$$\omega_l = \arg\min_{\omega} \sigma^2_{\hat{U}_{GM_{f_l}}} \tag{3.54}$$

The fusion weight is calculated independently for each grid cell, l, and defined as ω_l.

The choice of fusion weight ω does not affect the conservative fusion properties

from Theorem 3.3.2. In addition, the correct fusion rule is additive if the distributions of the height are uncorrelated from one node to another. This provides an opportunity to bound the maximum information loss as a result of the fusion rule (3.42) in the sufficient statistic set (3.16) for each variable:

$$Z_{f_{\text{uncorrelated}}} \triangleq Z_i + Z_j \tag{3.55}$$

$$Z_{\text{max loss bound}} \triangleq Z_{f_{\text{uncorrelated}}} - Z_f \tag{3.56}$$

$$= (1-\omega) \cdot Z_i + \omega \cdot Z_j \tag{3.57}$$

This novel information loss bound provides a means to track the quality of the fusion process. The information loss is computed for each individual grid cell and a information loss map is built as part of the distributed terrain mapping experiment. This provides an input to a planning process that guides agents to actively explore regions of the map where large information loss occurs in an effort to verify the map.

3.4 Laboratory Experiment

The data collection was performed in Cornell University's Autonomous Systems Laboratory (ASL) using the Pioneer P3-DX differential drive mobile robot from Mobile Robots Inc. shown in Figure 3.1. The P3-DX robot base is equipped with a custom built Mini-ITX computer running the Orca Robotics (Makarenko et al., 2007) software framework for sensor integration and control. The primary sensor for the terrain height estimation task is the compact Hokuyo URG-04X laser scanner which features a 240° field-of-view and angular resolution of 0.36°. The laser is pitched downward 45° and scans along the ground as the robot moves forward in a push-broom fashion.

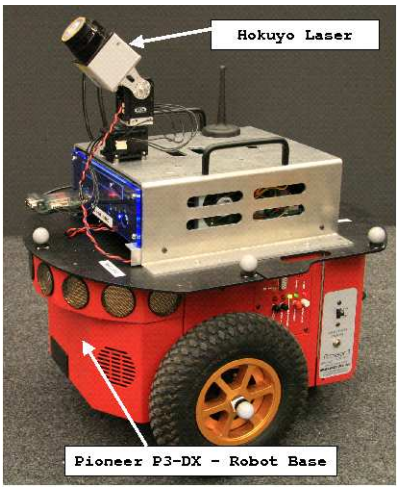

Figure 3.1: Mobile robot used in Cornell University's ASL equipped with Hokuyo laser scanner and on-board computer. The robot is used to explore and perform terrain mapping in the laboratory experiment.

The test environment is 15×8 meters and is instrumented with a Vicon MX+ precision tracking system that determines position and attitude of 3D objects (notice tracking markers on the mobile robot in Figure 3.1) in the test environment and is used to localize the robots. In certain areas of the test environment, precision tracking is not available and integrated odometry is used to localize the robots with a corresponding increase in pose uncertainty. The terrain features in the environment consist of boxes between 10 and 25 cm tall that are meant to simulate traffic cones or other similarly sized obstacles for a full-size traffic vehicle.

Eight robots are run in different paths around the environment for a 120 second data collection. The robots are run sequentially to avoid sensing another robot (dynamic terrain) during the terrain mapping. The paths of the robots are shown in Figure 3.3.

The paths of the robots allow each one to cover a different portion of the entire space with some overlap and the distributed terrain estimation algorithm combines

Figure 3.2: Laboratory environment for the distributed terrain mapping experiment covers 15×8 meters and contains boxes of different sizes. The experiment environment is representative of large obstacles relative to the robot on an otherwise flat terrain.

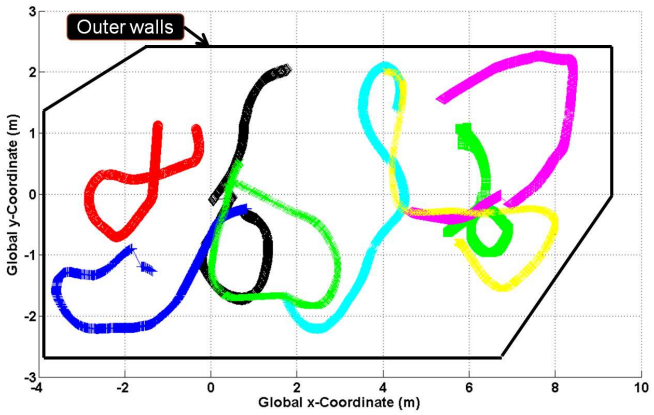

Figure 3.3: The overhead paths of the eight mobile robots exploring the environment for distributed terrain mapping experiment. The data collection interval is 120 seconds and different robots to explore different region of the environment with some overlap to demonstrate the benefit of the distributed terrain mapping techniques.

data from each sensor. The data from each sensor is run offline over a distributed computer network. Each computer is a standard PC and acts as an agent processing the recorded data. The communication between agents is performed using UDP over Ethernet. The UDP communication is lossy and communication is attempted after 5% of the local updates. This causes an asynchronous communication between agents and forces a distributed data fusion scheme that handles many local updates between communication. In addition, information from only a subset of the grid-cells are communicated at each attempted send. The grid-cells selected for communication are sorted according to length of time from last send and largest cumulative laser-to-grid-cell measurement association Z_p. The data from the local agents is saved in order to conduct important studies detailed below offline.

To analyze the data fusion problem, the total available information in the terrain map during the data collection is computed. The mixture-model based terrain estimation algorithm provides two convenient metrics for tracking the total available information. The first is the measurement-to-grid cell association probability Z_p^K (3.16), referred to as the Association Probability Metric (APM), and is used to compute the information content of any grid cell. For the terrain mapping problem the information content according to the APM at sensor node i in the grid cells observed C_O through time step k is:

$$\mathcal{I}_{p_i}^K \triangleq \sum_{l \in C_O} Z_{p_{i_l}}^K \tag{3.58}$$

The APM in node i is monotonically increasing, because there is no way to remove association probability from a given grid cell. This provides a finite and countable measure of the total information content in the sensor system at any given time and provides the ability to quantify accumulated information at each sensor node to assess the quality of the distributed data fusion algorithms. Therefore, the total

information available in the entire sensor network through time step k is:

$$\mathcal{I}_p^K \triangleq \sum_i \mathcal{I}_{p_i}^K \tag{3.59}$$

and the information available over the entire data collection $\mathcal{I}_p^{*K} \triangleq \mathcal{I}_p^{K_f}$ is given at the final time step K_f.

A second information theoretic method for computing the total information in the terrain map is the inverse of the cumulative entropy of the height distribution $p(U_l|\underline{r}^K, \underline{p}^K)$, referred to as the Cumulative Entropy Metric (CEM), in each grid cell l over time at sensor nodes i in the grid cells observed C_O through time step k:

$$\mathcal{I}_{e_i}^K \triangleq \frac{1}{\sum_{l \in C_O} H[p(U_l|\underline{r}^K, \underline{p}^K)]} \tag{3.60}$$

$$\approx \frac{1}{\sum_{l \in C_O} 0.5 \cdot \log\left(2\pi \cdot e \cdot \sigma^2_{\hat{U}_{GM_l}}\right)} \tag{3.61}$$

The CEM for the entire network is computed by first finding the centralized information sets according to (3.23) and the terrain map cell height variance according to (3.15) and then computing the CEM (3.61).

The approximation in the CEM (3.61) follows from computing the entropy of a Gaussian distribution resulting from computing the first and second moments of the Gaussian mixture. This is not the exact entropy of a Gaussian Mixture, but it is a conservative approximation (Huber et al., 2008). The summation is performed over each cell, because each cell is conditionally independent of one another. The CEM has two competing properties. First, as each cell is observed more, the variance in the cell decreases, and the summation in the denominator decreases. Second, as more cells are observed, the summation increases. Therefore, the desired behavior is to reach a stable equilibrium where all cells have been observed and have the smallest possible variance corresponding to the Cramer-Rao lower bound.

3.5 Experimental Results

The laboratory experiment is used to study several aspects of the distributed terrain mapping algorithm. The first analysis shows the terrain maps generated by each individual agent and presents a centralized solution as a benchmark. The next study examines the distributed terrain mapping approach using the channel filter over three different tree topologies to asses the quantitative information gain in terms of the Association Probability Metric (APM) and Cumulative Entropy Metric (CEM) at each node across the network. Another study is performed for the optimally connected network to evaluate the APM and CEM information gain with the generalized distributed terrain mapping technique using different fusion weight computation techniques. Finally, an evaluation of map dependent maximum information loss bound is presented for the optimally connected network..

The independent terrain maps generated for each agent after collecting data over the paths in Figure 3.3 are shown in Figure 3.4. The height displayed for the l^{th} cell in the centralized map is $\hat{U}_l \triangleq \hat{U}_{GM_l} + \sigma^2_{\hat{U}_{GM_l}}$. The bright red areas of the map around the edges are a set of walls surrounding the experiment area, the red areas inside the map are boxes placed in the environment and gray cells are unexplored by the agent.

The centralized terrain map is shown as the benchmark in Figure 3.5. The centralized terrain map is computed using the centralized solution information sets (3.23), which are computed from the individual node information sets (3.16). The outer walls exhibit smearing because of pose-uncertainty that occurs when the robot is localized using integrated wheel odometry. The green areas of the map represent areas that are safe for the robot to transit.

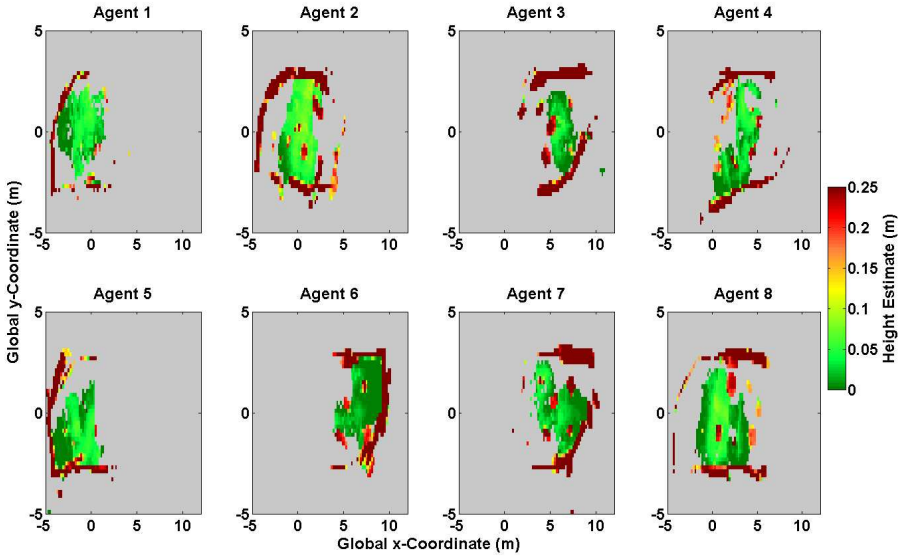

Figure 3.4: Final terrain map from each of the 8 robots without using the distributed terrain mapping algorithm. The maps show each robot only explores a subset of the entire area.

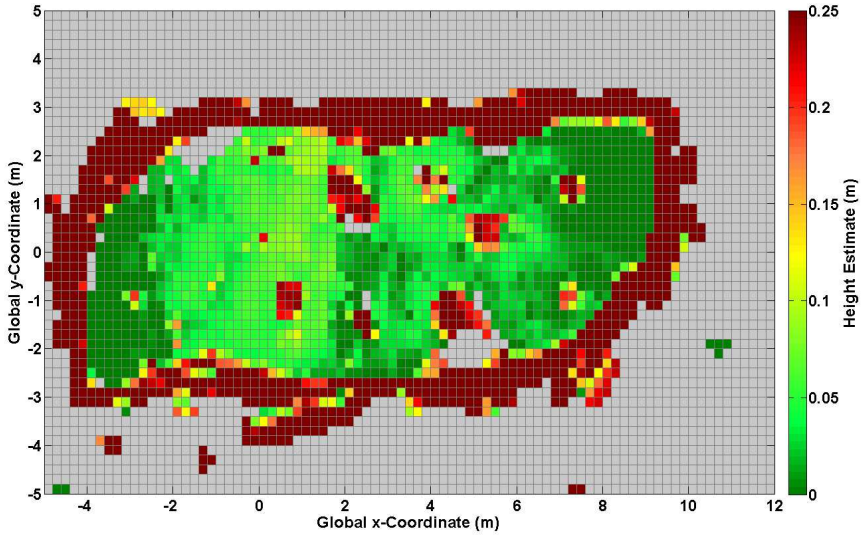

Figure 3.5: Final centralized terrain map from the 8 robots. The map shows the majority of the area is explored and the objects are recognizable, including the outer wall blurring demonstrate incorporating robot uncertainty in the terrain mapping technique. This map is used as a baseline for the distributed terrain mapping techniques.

For the centralized data fusion case, the information according to the APM and CEM available at each sensor node over the 120 second data collection is shown in Figure 3.6. The centralized solution APM and CEM provide a benchmark for the distributed data fusion algorithms and experiments. The top plot shows the Association Probability Metric (APM) for each node (3.58) and for the entire network (3.59). The monotonic increasing behavior of the metric is visible in the individual nodes and the centralized solution. The centralized solution APM provides a finite and countable measure of the total information content in the sensor network. Each sensor node contributes approximately equally to the APM of the entire network. The bottom plot shows the Cumulative Entropy Metric (CEM) (3.61) and that the centralized solution reaches the equilibrium, because the entire map is explored and the variance is reduced in each grid cell.

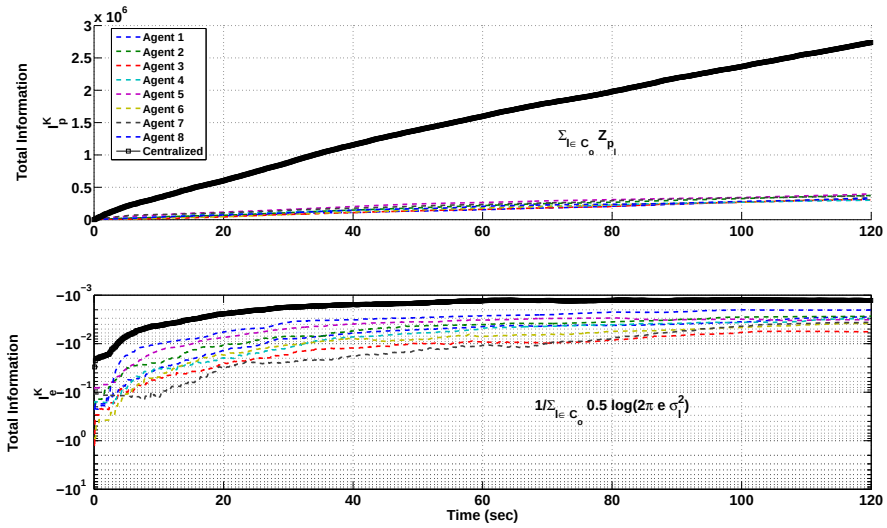

Figure 3.6: Information content as defined by APM (3.58) and CEM (3.61) for each sensor node without fusion and the centralized network solution (3.59). Each agent contributes nearly equally to the total information content.

3.5.1 Channel Filter Distributed Terrain Mapping on a Tree Topology

The distributed terrain mapping approach is studied using a chain, bridge and star tree topologies shown in Figure 3.7 to investigate the affect of the topology on information rate of gain at each node. For each tree topology, the channel filter ensures that the solution at each node converges to the globally optimal solution (Theorem 3.3.1). Said another way, the terrain map results for each topology eventually converge to the centralized solution shown in Figure 3.5. Each topology is selected for different properties. All of the topologies have the same order and size. The chain topology has nodes with the minimum degree; the star topology has the lowest graph diameter; and the bridge topology has properties between the other two. In any tree topology, a non-leaf node failure results in a disconnected graph. The chain topology is the least robust to node failure (fewest leafs); the star topology is least robust to a given node failure (the central node); and again the bridge topology is a balance between the two.

For the chain topology, the APM and CEM are shown in Figure 3.8 and the APM (top sub-plot) shows the delay across the network because the information must propagate along the maximum diameter graph. The leaf nodes (1 and 8) are lagging in information content compared to the other nodes, this demonstrates the delayed arrival of information content at the extreme nodes of the graph. At the end of the simulation, additional communication steps are required to propagate the latest collected local information across the entire network to reach the exact centralized solution; this was not done for clarity. The stair steps visible in the APM (top sub-plot) show the robustness to communication failure (failure to send or received) using the channel filter, because the local information continues to

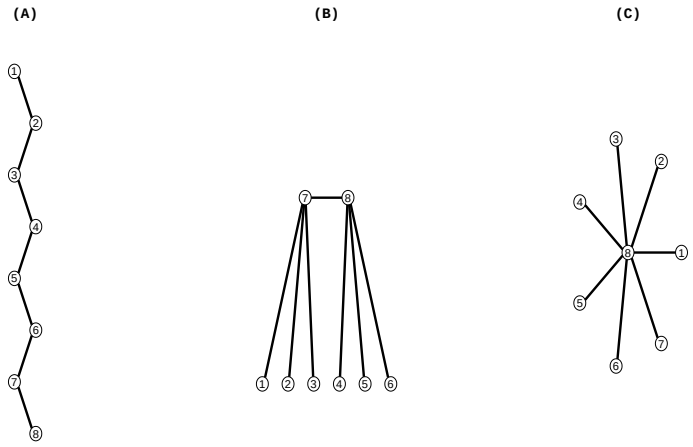

Figure 3.7: Different channel filter tree graph topologies; the chain (A), bridge (B), and star (C) topologies. The different topologies are used to investigate the impact of topology on information rate of gain at the different nodes in the network.

assimilate, until the remote information is received and the information content is updated without double counting. It is clear there is no double counting, because the APM (top sub-plot) does not exceed the centralized solution at any point. The CEM (bottom sub-plot) shows that each node converges to a similar plateau as the centralized solution, but the solutions are not exact because of delay in the network.

The star topology shows similar information consistency and convergence to the centralized solution as the chain topology and is shown in Figure 3.9. However the central node (Agent 8) always has the most information (according to the APM) of any node because all information must flow through node 8 before being sent onto any other node. The final solution is closer to the centralized solution than the bridge topology, because of the reduced diameter of the network. The amount of missing information at any node (difference between the centralized and

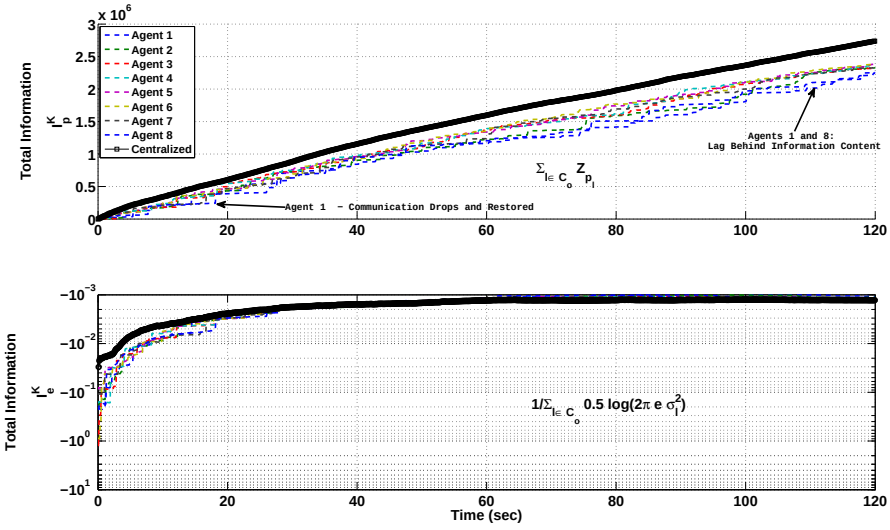

Figure 3.8: APM and CEM for the channel filter chain topology at each individual node is shown along with the centralized solution. The information is delayed propagating across the network as the leaf nodes (Agents 1 and 8) trail the information content of the non-leaf nodes from the APM (top sub-plot).

node solution) is reduced in the star topology compared to the chain, because the maximum eccentricity of the graph is 2. This implies that all information in the sensor network is within 2 communication steps from any other node.

The bridge topology exhibits hybrid results compared to the chain and star topologies and is shown in Figure 3.10. The bridge topology has no node with significantly more or less information content (according to the APM) than the other nodes as observed in the chain and star topologies, but the information content at the central nodes (Agents 7 and 8) is slightly higher than the remaining nodes.

The different topologies have slightly different characteristics in the information gain according to the APM and CEM achieved at each node depending on the topology. Unfortunately, the tree topology and channel filter have several lim-

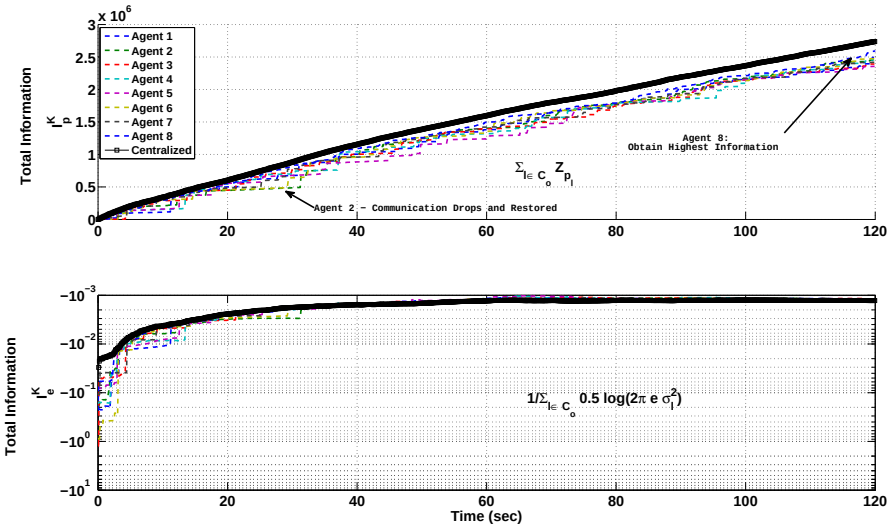

Figure 3.9: APM and CEM for the channel filter star topology at each individual node is shown along with the centralized solution. The results show that information at the central node (Agent 8) has the most information of any node from the APM (top sub-plot).

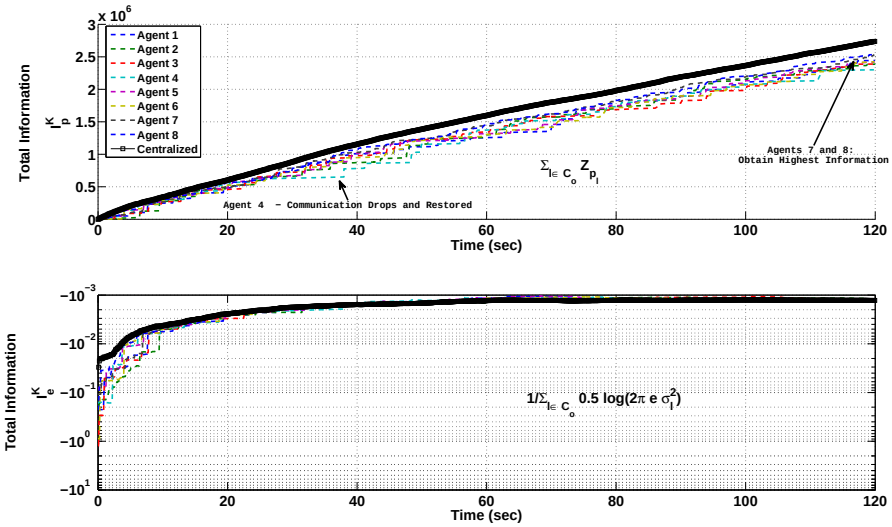

Figure 3.10: APM and CEM for the channel filter bridge topology at each individual node is shown along with the centralized solution. The results show that information at the central nodes (Agent 7) and (Agent 8) have the most information from the APM (top sub-plot).

itations. The graph becomes disconnected with a single node or link failure. In addition, the network must enforce a tree topology, which make reconfiguration difficult. The Distributed Bellman-Ford algorithm (Awerbuch et al., 1994) can be used to discover cycles in the graph and break the links artificially, but because the channel filter cannot recover from double counted information, it must re-initialize on a reconfiguration. This implies that either performing generalized fusion for a step or taking one node estimate instead of the other is required. These issues are all formally handled with the generalized fusion approach presented next.

3.5.2 Generalized Fusion on an Optimal Connected Topology

To study the application of the generalized distributed terrain mapping technique, the laboratory data is analyzed with the sensor network configured in an arbitrary connected topology. This enables the examination of the information content according to the APM and CEM when applying the generalized fusion techniques. Additionally, the maximum information loss bound (using the APM) is examined when applying the generalized fusion technique.

In the generalized fusion approach, as applied to the experiment, it is setup that the nodes are connected according to the topology with maximum robustness (Dekker and Colbert, 2004). The topology is symmetric, has equal node connectivity and link connectivity that is equal to the minimum degree, and each node is a central vertex of the graph. The topology shown in Figure 3.11 consists of 12 links: each node has degree 3 and requires a minimum of 3 nodes to fail to become disconnected. By comparison, any of the tree topologies required for the channel

filter require just 1 node or link to fail to become disconnected, unless the node is a leaf node.

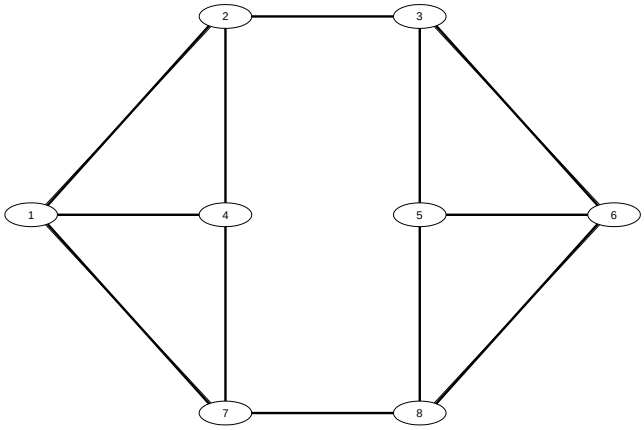

Figure 3.11: Network topology for the generalized distributed terrain mapping technique. The graph is optimal for 8 nodes and has an equivalent node and link connectivity and requires 3 nodes to fail to become disconnected.

The first analysis performed with the generalized data fusion approach is to examine the influence of how the fusion weight ω_l is computed (3.52)-(3.54) for each grid cell. In the first approach, the weight is fixed $\omega_l = 0.5$. This is the simplest approach and requires no additional calculations. The resulting terrain maps for each node are shown in Figure 3.12. The results are qualitatively similar to the optimal centralized solution and each node achieves a representation of the full-environment. Quantitatively, however, there are differences from the optimal solution, as shown in Figure 3.13. The APM, $\mathcal{I}_p$, in the top sub-plot, is approximately equivalent at each node to the total content available without fusion. This is primarily because of discounting the exclusive information during fusion. The fusion rule is in fact conservative, because the APM of individual nodes never ex-

ceeds the centralized solution. The CEM, $\mathcal{I}_e$, in the bottom sub-plot, shows that the maps are qualitatively the same because nearly all of the information that is observed in a given cell is accounted for in the fused solution.

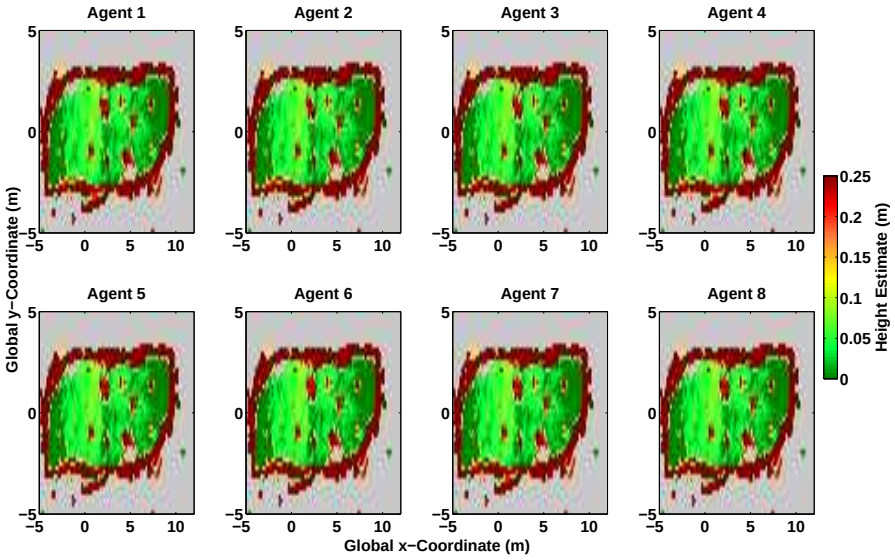

Figure 3.12: The final distributed terrain map at each of the 8 nodes when using a fixed $\omega = 0.5$ fusion weight (3.52) with the optimally connected topology in Figure 3.11. The results show the terrain maps at each node are qualitatively similar to the optimal centralized fusion Figure 3.5 providing a more robust method for estimating the distributed terrain map.

The next approach computes the fusion weight based on the cell-association probability of the corresponding nodes (3.53). This approach has a nice property that if a given node has not observed a cell, that node simply takes the result from the neighboring cell without any special computations because $\omega = 0$. The resultant terrain maps are not qualitatively different from Figure 3.12, so they are not repeated here. Quantitatively, the results of the information content according to the APM and CEM are shown in Figure 3.14. The APM (top sub-plot) shows the rule is conservative and an interesting quality about the APM is that it rewards nodes that are not exploring the same area of the world. The CEM (bottom sub-

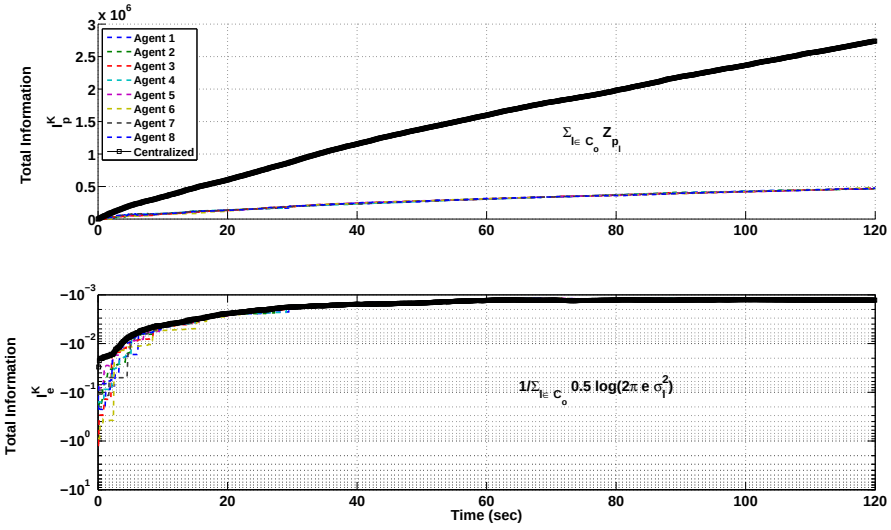

Figure 3.13: The APM and CEM at each of the 8 nodes when using a fixed $\omega = 0.5$ fusion weight (3.52) with the optimally connected topology in Figure 3.11 along with the centralized result. The results shows, by the APM (top sub-plot), that the fusion result is lossy, but the CEM (bottom sub-plot) shows each node approaches the centralized solution in total cells observed and approximate variance of the height estimate in those cells.

plot) converges to the centralized solution, because there is an estimate of the height and corresponding uncertainty is obtained in each grid-cell over the entire map that matches the centralized solution.

The final approach for computing the fusion weight is to minimize the Gaussian Mixture covariance for each grid cell (3.54). This can be computed in closed form, because the Gaussian Mixture covariance (3.15) is a scalar. Again, the resulting terrain maps at each node are qualitatively similar to the centralized solution are are not shown here. The APM and CEM are shown in Figure 3.15. The APM (top sub-plot) is equivalent to the fixed weight generalized fusion solution (Figure 3.13). The interesting artifact of this fusion rule is observed as the CEM (bottom sub-plot) begins to reach an equilibrium around 50 seconds. The subsequent fusion

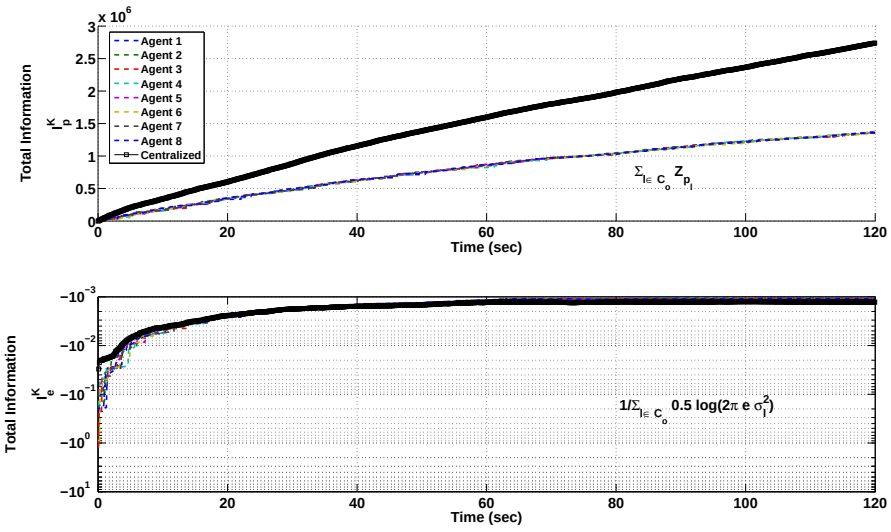

Figure 3.14: The APM and CEM at each of the 8 nodes when using a probabilistic fusion weight (3.53) with the optimally connected topology in Figure 3.11 along with the centralized result. The results shows that the APM (top-subplot) is increased compared to a fixed weight Figure 3.13, because information is not discounted when there is no conflict of observations.

steps result in reducing the variance of the resulting height estimate after fusion. This causes the denominator in (3.54) to decrease in magnitude, which causes a subsequent increase in magnitude (more negative) of the entropy metric. This suggests that while not receiving all of the information, the conservative fusion rule can achieve reduced variance estimates for the terrain height in each grid cell.

3.5.3 Maximum Information Loss Bound

After applying the generalized distributed terrain mapping algorithm, the maximum information loss bound is computed for each of the agents across the entire map using the Association Probability Metric (APM). The study of the maximum information loss bound evaluates and demonstrates the location dependent nature

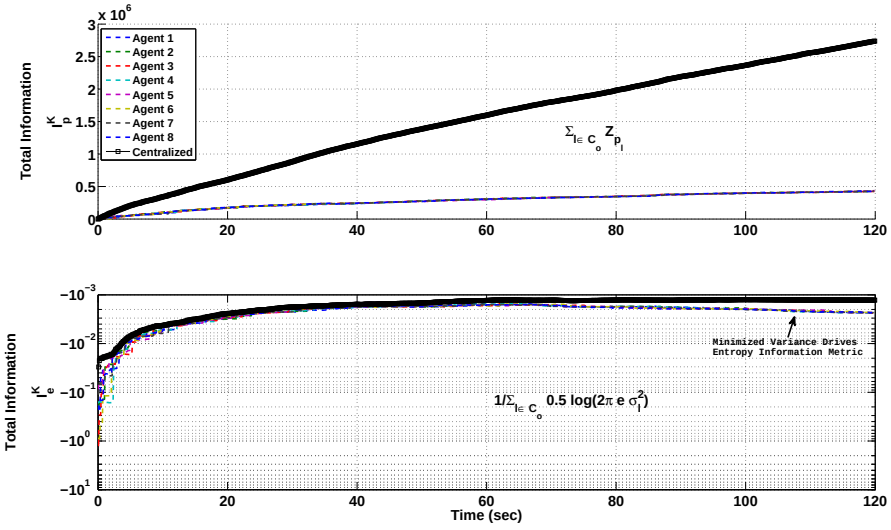

Figure 3.15: The APM and CEM at each of the 8 nodes when using a minimize variance fusion weight (3.54) with the optimally connected topology in Figure 3.11 along with the centralized result. The results show the benefit of the minimum variance fusion weight in the CEM (bottom sub-plot) after approaching steady state, as the covariance is driven down the CEM is conversely made larger in magnitude.

of the information loss. An analysis to of the cumulative information loss bound shows that the bound is not tight. However, an empirical cumulative probability distribution across all of the agents and map cells of the maximum information loss is computed to show that while the bound is conservative, it is small for a majority of cells and therefore could be used to balance exploration and verification of the few cells with potentially large information loss.

The maximum information loss bound, equation (3.57) is evaluated over each grid cell in the map and cumulatively over the entire map over time. The probabilistic weighting fusion approach (3.53) is used in this study, because if one node observes a cell and the other has not, the maximum loss is correctly computed as 0. The cells that are observed the most have the highest maximum informa-

tion loss. The plot of the maximum information loss for each grid cell is shown in Figure 3.16 for each sensor node. Each node converges to the same estimate of the maximum information loss, because nodes share and fuse the information similarly. The areas in red have the highest potential loss and represent the areas that are actually observed by the most sensor nodes. This presents an interesting dichotomy: the most information is potentially missing, but since the cells have been observed the most the variance in the height estimate is driven to the lowest level. These maximum information loss map and the uncertainty in the terrain map can be used for planning to balance verification and exploration.

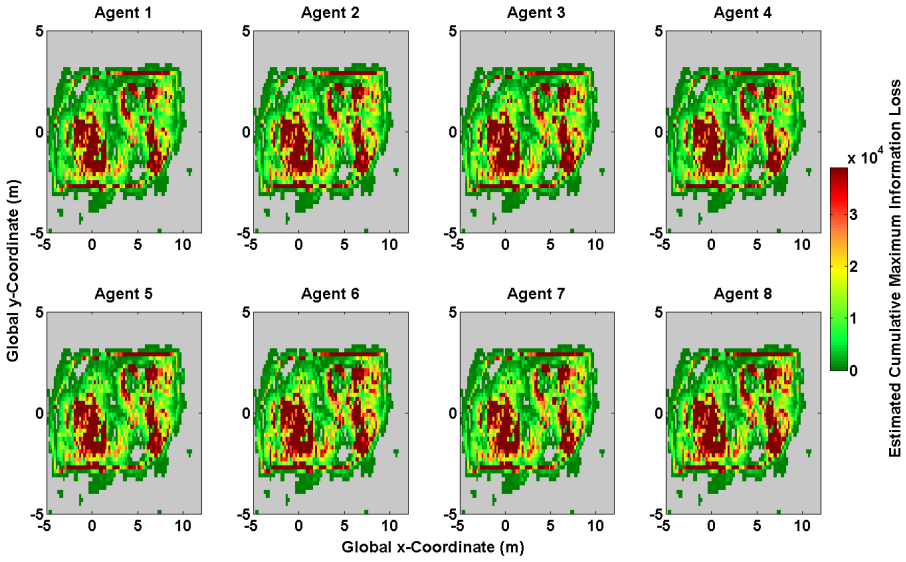

Figure 3.16: The estimated maximum information loss map at each of the 8 nodes when using a probabilistic fusion weight (3.53) with the optimally connected topology in Figure 3.11. The results show grid-cells commonly observed by multiple agents (dark red) have the maximum information loss bound under the generalized fusion scheme, because there is the greatest number of fusion events discounting exclusive information.

The cumulative information loss according to the APM over the entire map is shown in Figure 3.17 and is computed from summing the maximum information

loss bound (3.57) that is computed after each fusion step. Figure 3.17 demonstrates that the maximum loss information bound is not tight (since it far exceeds the cumulative information in the centralized solution) and is actually growing increasingly further from the true value over time. This is because each estimate of maximum information loss is an overestimate. To make best use of the cumulative information loss bound, the bound could be used as a synchronizer to trigger reset of the terrain map between any two neighboring nodes to be equal to the map with the lowest variance in each grid cell. This approach would allow the maximum information loss bound to be reset to zero and avoid accumulating increasingly conservative estimates of the bound.

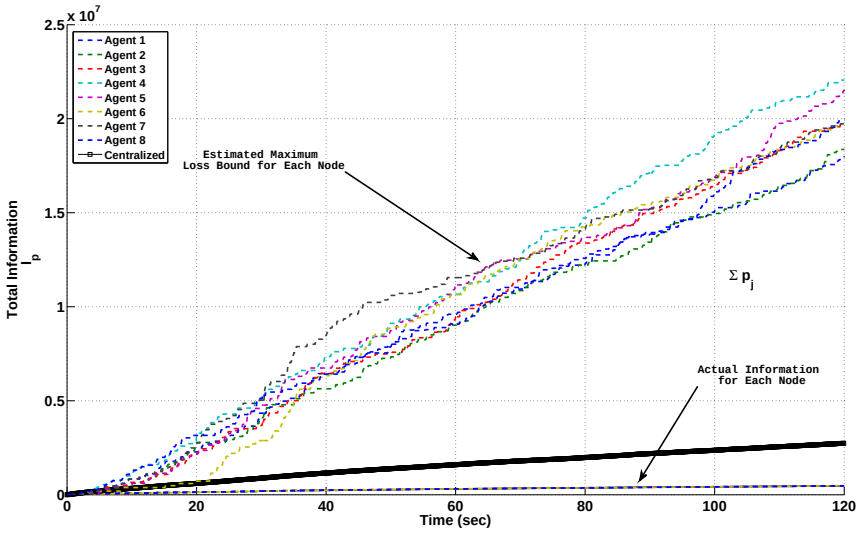

Figure 3.17: The cumulative estimated information loss bound (3.57) using the APM at each of the 8 nodes when using a probabilistic weight (3.53) with the optimally connected topology in Figure 3.11 along with actual information content (according to the APM) of the individual nodes and centralized result. The results show that the cumulative bound is not tight as it grossly exceeds the actual cumulative centralized information.

A nice feature of the maximum information loss is that it can be computed for each grid-cell. The loose bound for the cumulative maximum information loss can

be used as a reset trigger, but individual grid cell loss bounds can be used to guide exploration. Figure 3.18 shows the cumulative distribution function approximation for the maximum information loss across every grid cell of a particular node in the generalized fusion. The interesting part is that the majority of the cells have a very small maximum information loss and can be safely regarded as having all the information available in the entire network. The small number of cells on the tail of the distribution can therefore become the focus of verification efforts.

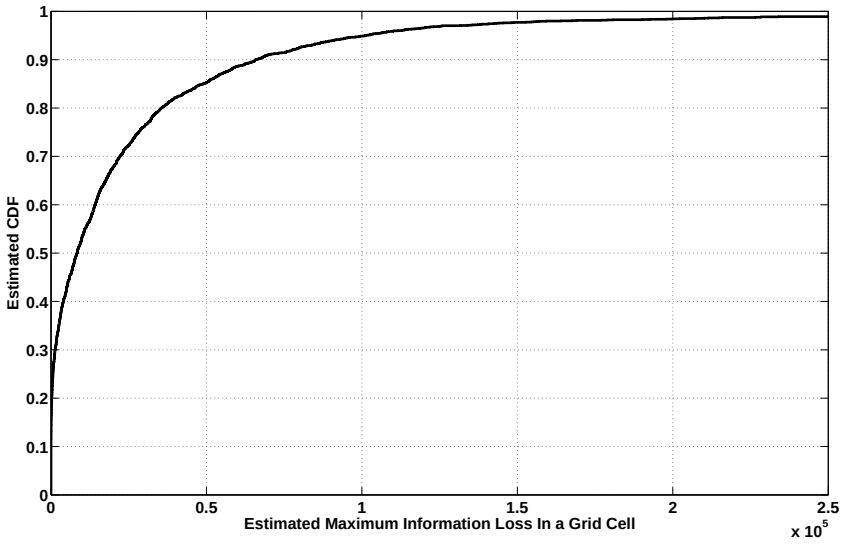

Figure 3.18: The estimated CDF, across all 8 nodes of the maximum information loss when using a probabilistic weight (3.53) with the optimally connected topology in Figure 3.11. The results show that the bound is generally small for individual grid cells, but has a long tail due to cells that are repeatedly fused and discount exclusive information, leading to a potentially large information loss.

3.6 Conclusions

The grid-based terrain mapping algorithm that maintains a probabilistic accuracy assessment of the height estimate in each grid-cell is successfully developed and demonstrated in laboratory experiment in the distributed data fusion paradigm. Two approaches were developed for distributed data fusion using sufficient statistics of the cell height Gaussian Mixture computed for each grid-cell in the map. First, the exact solution is recovered through tracking of the common information in the sufficient statistic through use of the channel filter when the network is tree connected. The chain, bridge and star topologies were shown to have similar, but slightly different information gain characteristics over according to the Association Probability Metric (APM) and Cumulative Entropy Metric (CEM) over the sensor network. Each topology quantitatively converges in terms of both information metrics to the global optimal solution. The difficulties in terms of network robustness and reconfiguration ease motivate the use of arbitrary connected networks.

Second, an approach for fusion in arbitrary networks was derived that uses weighted combination of the sufficient statistic set of the Gaussian Mixture for each grid cell. The fusion rule operates on the sufficient statistics of the Gaussian Mixture height distribution and provides a unique resultant Gaussian Mixture mean and covariance. The fusion rule is shown to be quantitatively consistent, but conservative in terms of the APM and CEM using an optimally connected network.

Three different techniques for computing the fusion weight are presented for the generalized fusion. The resultant maps are qualitatively similar to the optimal centralized solution regardless of the weight calculation technique. The fixed weight and minimum variance weight computations are shown to discount the most

information according to the APM in the network, but still have convergence to the global solution in terms of the CEM. The probabilistic weight computation conserves the most information according to the APM in the network compared to the other techniques, which encourages sensor nodes to explore disjoint areas of the map. The conservative fusion rule is shown to discount information available in the sensor network and a bound of the maximum information loss for each grid cell is derived.

The cumulative maximum information loss is shown to be a loose bound, because it is an accumulation of repeated over-estimates. The maximum loss is shown on the terrain map and demonstrates the dichotomy that repeated fusion of the same grid-cells leads to reduced variance, but increases the apparent information loss. The cumulative distribution of the maximum information loss across cells shows heavy tails and motivates the ability to plan paths in the environment based on verification of cells with potentially high loss and exploration and previously unobserved cells.

Acknowledgments

The authors would like to thank Isaac Miller and the members of the Cornell University Autonomous Systems Laboratory who made data collection possible. This work is supported under the Northrop Grumman Electronic Systems Scholars Program.

CHAPTER 4

SEGMENTATION OF DENSE RANGE INFORMATION IN COMPLEX URBAN SCENES

Abstract

In this paper, an algorithm to segment 3D points in dense range maps generated from the fusion of a single optical camera and a multiple emitter/detector laser range finder is presented. The camera image and laser range data are fused using a Markov Random Field to estimate a 3D point corresponding to each image pixel. The textured 3D dense point cloud is segmented based on evidence of a boundary between regions of the textured point cloud. Clusters are discriminated based on Euclidean distance, pixel intensity and estimated surface normal using a fast, deterministic and near linear-time segmentation algorithm. The algorithm is demonstrated on data collected with the Cornell University DARPA Urban Challenge vehicle. Performance of the proposed dense segmentation routine is evaluated in a complex urban environment and compared to segmentation of the sparse point cloud. Results demonstrate the effectiveness of the dense segmentation algorithm to avoid over-segmentation better than incorporating color and surface normal data in the sparse point cloud.

4.1 Introduction

Large scale autonomous vehicles operating in human-populated areas are required to accurately perceive their environment to make safe and robust decisions. One challenge in perception is the accurate identification of objects in the environment. Identification of objects requires accuracy and robustness to maintain target track, avoid collisions, minimize phantom objects, and predict future intent of objects (Fletcher et al., 2008). The autonomous vehicle must rely on its on-board sensors to perceive the environment and data processing techniques to identify objects of interest.

The available sensors on many autonomous vehicles include laser range finders and high precision optical cameras. Most autonomous vehicles segment sensor data because of the large amount of data, but have done so independently with each sensor (Miller et al., sion). Miller et al. (Miller et al., 2008a) describe the existing algorithm for performing laser range finder point clustering on the Cornell University DARPA Urban Challenge Vehicle. In particular, the algorithm creates ground models from lidar data to identify drivable paths and reason about objects in the environment (Miller et al., 2008a). This algorithm fails to utilize the richness of information available from optical sensors. For example, a common mistake occurs when the vehicle determines the road is blocked, when in fact the car is moving along a road with a large change in pitch. One solution to these problems is data fusion, which has been shown to increase robustness for urban based perception (Miller et al., 2008a). Here, it is proposed to fuse lidar and vision to create a dense range map with more information than one sensor alone. This dense range map is segmented and can be used for ground detection or as a pre-processing step for object tracking. This paper is concerned with segmentation of point clouds

that have not undergone any temporal filtering at a higher level to ensure uncorrelated measurements that are passed to higher level functions such as stable cluster tracking (Miller et al., sion).

Diebel and Thrun (Diebel and Thrun, 2006) model the correlations between the camera image and laser range data using a Markov Random Field (MRF). Gould et al. (Gould et al., 2008) relax the fronto-parallel planes assumption in Diebel and also use a Huber cost instead of the l_2 cost; better results are claimed because of the first-order model. Harrison and Newman (Harrison and Newman, 2009) add an extrapolation capability to Diebel's MRF approach and were the first to find the MRF maximum a posteriori (MAP) estimate with a direct method (using Matlab®) instead of an iterative method. Andreasson et al. (Andreasson et al., 2007) identify five methods for interpolation of 3D laser scans on pixel density that are outside the MRF framework and introduce a confidence measure. However, mixed performance results of the different methods are seen when comparing to Diebel's MRF. The algorithm presented here uses Diebel's original MRF formulation to produce a textured 3D dense point cloud, but uses an efficient direct solver to move towards real-time performance. In this work, real-time performance is the goal because the entire frame of laser data overlapping the image is captured in near simultaneity with a multiple emitter/detector laser range finder, enabling dense ranging in dynamic scenes.

After the lidar and camera data are fused, the next step is segmentation of the textured 3D dense point cloud. A segmentation algorithm that is deterministic, runs efficiently, and incorporates information about the 3D geometry and pixel intensity is desired. Besl and Jain (Besl and Jain, 1988) introduce segmentation of point clouds using variable order functions to model the surface and an iterative

refinement of groups, but the algorithm presented here use a non-iterative and non-parametric segmentation technique. Gachter et al. (Gachter et al., 2006) present an algorithm for extracting planes by first extracting lines segments in an indoor environment. However, the algorithm presented here does not make assumptions about the environment. Rusu et al. (Rusu et al., 2007) demonstrate segmentation of lidar returned in an indoor environment based on surface normal and curvature estimates, but the algorithm also relies on the structure of the indoor environment and sequentially registered 3D point clouds. Klasing et al. (Klasing et al., 2008) present a deterministic radially bounded segmentation scheme. For performance comparison, the algorithm in (Klasing et al., 2008) is evaluated on the data collected in this study and the algorithm is modified to incorporate information about pixel intensity and estimated surface normal. Lim and Suter (Lim and Suter, 2008) use Conditional Random Fields (CRF) to segment point clouds of terrestrial features, but the computations require accurate registration of sequential scans and several hours of run-time. Felzenszwalb and Huttenlocher (Felzenszwalb and Huttenlocher, 2004) introduce a fast greedy predicate based segmentation algorithm for computer vision. The algorithm is extended to arbitrary points clouds, instead of images. The algorithm in (Felzenszwalb and Huttenlocher, 2004) is used here because of its fast, nearly linear run-time, deterministic segmentation, global guarantees on segment size, and ease of incorporating pixel intensity and 3D geometry into the segmenter predicate.

The novelty of this paper is the ability to efficiently and accurately segment dense range maps produced by the fusion of camera and range data in a complex urban environment in near real-time. The algorithm is demonstrated on experimental data recorded from the Cornell University DARPA Urban Challenge vehicle. Section 4.2 describes the fusion of the camera and laser range finder data,

segmentation of the textured dense point cloud and an efficient implementation. Section 4.3 shows the algorithm applied to experimental data collected in an urban environment from a moving platform. Finally, Section 4.4 summarizes with conclusions demonstrating the success of the algorithm to segment dense point clouds in a complex urban environment.

4.2 Segmentation of Dense Range Maps

The 3D textured point cloud segmentation algorithm begins by fusing camera images with a multiple emitter/detector laser range finder. Multiple emitter/detectors enable the near simultaneous capture of an entire sparse 3D point cloud that overlays the camera image (Figure 4.1). The camera and laser range finder data are fused to produce a textured dense point cloud. A graph $G(V, E)$ is formed from the dense point cloud where the vertices (V) are the 3D points and edges (E) connect vertices. An edge cost function is computed for all edges in the graph. The segmentation algorithm proceeds linearly over the edges and joins vertices into segments if the edge cost falls below an adaptive threshold. The final segments are available for object tracking or other high-level processing.

4.2.1 Fusion of Camera and Laser Range Finder Data

The fusion of camera and laser range finder data is accomplished by using an MRF to model the correlations between changes in the pixel intensity and depth changes in the corresponding laser range finder (Diebel and Thrun, 2006). The MRF defined to interpolate the sparse laser range finder data at the image pixel

Figure 4.1: Color camera image shows four pedestrians and another vehicle on a bridge near Cornell University's campus.

resolution is:

$$p(y|x,z) = \frac{1}{Z}\exp\left\{-\frac{1}{2}\left(\Phi+\Psi\right)\right\} \tag{4.1}$$

$$\Psi = \sum_{i\in\mathcal{L}} k[y_i - z_i]^2$$

$$\Phi = \sum_{i}\sum_{j\in\mathcal{N}_i} w_{ij}[y_i - y_j]^2$$

$$w_{ij} = \exp\left(-c\cdot u_{ij}\right)$$

$$u_{ij} = \|x_i - x_j\|^2$$

where y is the desired interpolated 3D point cloud, x is the image intensity data, and z is the sparse laser range finder returns. Additionally, k and c are tuning constants, the set $\mathcal{L}$ corresponds to the pixels that have aligned laser range data, and $\mathcal{N}_i$ corresponds to the cardinal neighbors of pixel i in the image.

The first step in interpolating the laser range finder data (range r, yaw θ, and pitch ϕ) is to align the sparse laser range finder data to the image:

$$\vec{v}_i = C(x,y,f)\cdot T(\vec{p})\cdot f(r,\theta,\phi) \tag{4.2}$$

$$u = \lfloor \vec{v}_{i_x}/\vec{v}_{i_z} \rfloor$$

$$v = \left\lfloor \vec{v}_{i_y}/\vec{v}_{i_z} \right\rfloor$$

where $f(\cdot)$ is the transform from spherical to Cartesian space, $T(\vec{p})$ is the transformation matrix from the laser range finder to camera coordinates, parameterized by a translation and rotation in $\vec{p}$, $C(x, y, f)$ is the 3D to 2D camera projection matrix and u and v are the corresponding width and height image pixels, as shown in Figure 4.2. The calibration of the camera to laser range finder transformation was performed by placing colored IR reflective markers at various points in a stationary image and recording laser range finder points and determining the transformation that aligned the high-intensity IR returns with the color markers in the image.

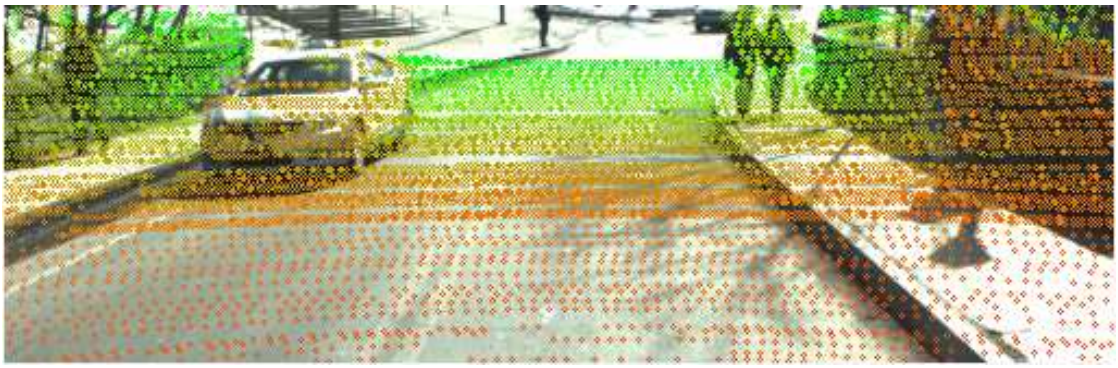

Figure 4.2: Aligned sparse laser range finder data (falsely colored points) on an image of an urban scene.

The desired 3D point cloud y in (4.1) is found by determining the maximum a posteriori estimate of $p(y|x, z)$, which is equivalent to minimizing the negative exponential cost function $J(y) = 0.5(\Phi + \Psi)$. The minimization of $J(y)$ is cast as an unconstrained quadratic program $f(y) = \frac{1}{2}y^T Ay - b^T y$, where $A = (A_\phi + A_\psi) \in \Re^{n_{\text{pix}} \times n_{\text{pix}}}$ is a large sparse symmetric positive definite matrix (n_{pix} is the number of pixels in the image) with $b = A_\psi^T z$. The solution y is found by solving the linear system $Ay = b$. The large sparse linear system is solved via a direct method using the Intel®Math Kernel Library DSS sparse matrix solver (Intel®, 2011) that takes advantage of the banded matrix structure to produce dense 3D point clouds efficiently. The final dense textured 3D point cloud is shown in Figure 4.3

and is passed on to the segmenter.

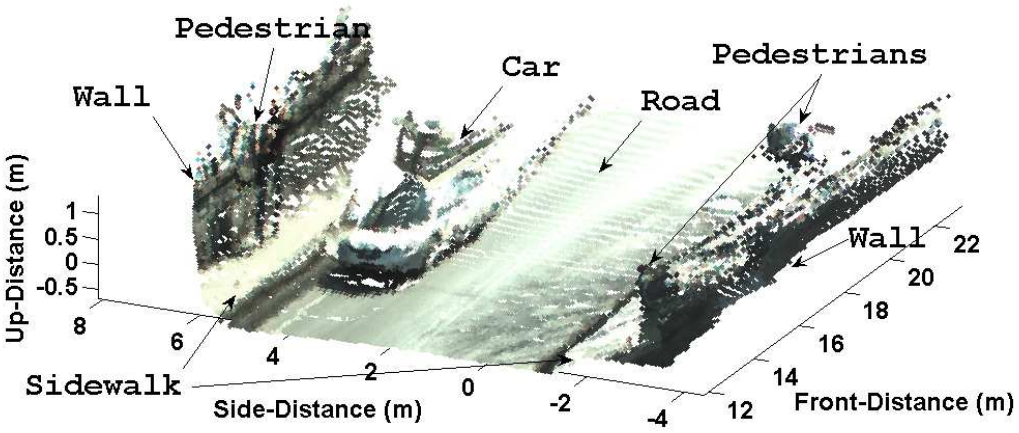

Figure 4.3: Interpolated dense 3D point cloud textured according to the original image.

4.2.2 Segmentation of Textured Dense Point Cloud

A predicate based segmentation routine (Felzenszwalb and Huttenlocher, 2004) is utilized to segment the 3D points in the textured dense point cloud. Unlike the algorithm in (Felzenszwalb and Huttenlocher, 2004), the segmentation routine uses 3D geometry, pixel intensity and estimated surface normal at each point in the dense cloud to discern segments. The segmentation algorithm operates on an undirected graph $G(V, E)$, where the vertices are the 3D points and the edges correspond to connections between vertices. The algorithm is initialized by computing the edge cost ω_{ij} between neighbors i and j in the graph, which is defined as a weighted combination of Euclidean distance, pixel intensity difference and the estimated surface normal divergence:

$$\omega_{ij} = k_e \cdot \|\vec{v}_i - \vec{v}_j\|^2 + k_i \cdot \|x_i - x_j\|^2 + k_N \cdot (1.0 - \vec{N}_i^T \vec{N}_j) \tag{4.3}$$

where $\vec{N}$ is the estimated surface normal at the 3D point, and k_e, k_i and k_N are the relative weights of the Euclidean distance between points, the pixel intensity difference and surface normal distance. The relative weights are hand-tuned for this implementation; the results presented below were insensitive to small changes in weights, but could be determined via cross-validation. Note that the lower edge weight implies that it is easier to merge two clusters.

The algorithm is initialized with each vertex (3D point) as a disjoint set and proceeds linearly along edges merging vertices into a cluster if the edge weight is below the cluster threshold η_i. The key to the algorithm is the update of the cluster threshold as vertices are joined. This has the effect that smaller clusters require greater evidence of a boundary to avoid being merged. The benefits of this segmentation routine are that it runs in near linear time to the number of pixels, is deterministic, and guarantees clusters of a certain size based on the tuning parameter σ_k listed in Algorithm 1.

The primary difference between the approach in (Klasing et al., 2008) and Algorithm 1 is the traversal over the edges, instead of vertices, and the adaptive threshold as a function of cluster size. There are many possible choices of edges (E) between the vertices (V) on the graph $G(V, E)$ when the vertices are 3D points in space. One approach uses $k-$nearest neighbor, where edges are created between the closest k vertices using any desired cost function. Another approach is to choose vertices that lie within a radial bound of each other, similar to (Klasing et al., 2008). The algorithm presented here begins by creating a dense range map, where a 3D point is estimated for each pixel in the underlying image. As a result, it is possible to create edges between 3D points corresponding to neighboring pixels in the image. It is important to note that two vertices not connected with an edge

Algorithm 1: Minimal Spanning Tree Segmentation (Felzenszwalb and Huttenlocher, 2004)

Compute edge weights according to (4.3)
Initialize cluster thresholds: $\eta_i = 1/\sigma_k, \forall i \in V$
for all Edges in Graph $G(V, E)$ **do**
    Extract set A and B along the edge
    if ROOT{A} != ROOT{B} **then**
        if $\omega_{ij} < \eta_i$ **then**
            $A \leftarrow A \bigcup B$
            $\eta_i = \eta_i + \text{SIZE}(A)/\sigma_k$
        end if
    end if
end for

may end up in the same cluster through intermediate vertices and two vertices connected with an edge will not necessarily be in the same cluster.

4.2.3 Surface Normal Estimation

The surface normal is computed using Principle Component Analysis (PCA)(Klasing et al., 2009). Similar to (Klasing et al., 2009), the analysis presented here found the PCA method performed better than the area averaging method (Klasing et al., 2009) to estimate the surface normal. The surrounding points used to determine the surface normal can be determined from the N-nearest neighbors, where nearest is a Euclidean norm. The estimated surface normal is ambiguous in terms of sign; to account for this ambiguity the dot-product between

estimated surface normals is repeated using the negative estimated surface normal of one of the vectors and the minimum result of the term multiplied by k_N in (4.3) is selected. For the sparse data set, if all nearest neighbors lie on a single laser scan line, the surface normal estimate is invalid. Therefore, these points have no valid surface normal estimate and k_N in (4.3) is set to zero. For the dense data set, the neighboring points used to estimate the surface normal are not from the same laser scan line, therefore all points have a valid surface normal.

4.3 Experimental Results

4.3.1 Autonomous Vehicle Testbed

Experimental data were collected for evaluation of the textured dense 3D point cloud segmentation routine using Cornell University's DARPA Urban Challenge vehicle (Miller et al., 2008a), shown in Figure 4.4. The optical camera mounted along the center roofline is a color Basler A622f with a 4.8mm lens and a 30-deg horizontal and 30-deg vertical field of view. The Basler camera produces 1280 x 1024 pixel images at 16Hz. The multiple emitter/detector laser range finder located on top of the vehicle is the Velodyne HDL-64E. The Velodyne scanner receives 64 vertical lines over 26.8 degrees of laser range finder data over 360 degrees at 15 Hz with a maximum detection range of 50 m. The scan rate of the laser range finder means all the laser range data overlapping the image is collected in 5.56 msec, at an interval of 61 msec. The portion of the image with overlapping laser range data is 1280 x 330 pixels, corresponding to an estimation of $422,400$ 3D points in the dense range map, which is more than 60 times greater than the number of 3D

Figure 4.4: Cornell University's autonomous Chevrolet Tahoe, equipped with a high precision optical camera and a multiple emitter/detector laser range finder.

points in the original laser range scan overlapping the image.

4.3.2 Urban Environment Data Collection

The algorithm was run on data collected near Cornell University in Ithaca, NY with the vehicle under human control on a busy afternoon. The scenes observed include moving cars, pedestrians, parked cars and standard urban features such as buildings, newspaper stands, and street signs. The initial sequence of analyzed scenes correspond to the vehicle passing by pedestrians on the side of the road and another vehicle while on a bridge (Figure 4.1). As a performance metric, the object-level consistency error (OCE) introduced by Polak et al. (Polak et al., 2009) is used to compare the segmented images with hand-labeled truth. The OCE metric weights segmentation performance as a function of the segment size. The OCE

heavily penalizes over-segmentation, so the global consistency error (GCE) and local consistency error (LCE) introduced by Martin et al. (Martin et al., 2001) are also shown. These metrics favor segmentation that is intuitively reasonable. There are 13 frames analyzed for segmentation performance, 9 proceeding and 3 following the frame in Figure 4.1. The textured dense 3D point cloud segmentation routine described in Section 4.2 is run and performance summarized. To demonstrate the importance of densifying the sparse 3D point cloud, the radially-bounded nearest neighbor (RBNN) algorithm from (Klasing et al., 2008) and a modification of the RBNN algorithm to incorporate pixel intensity and estimated surface normals are used to segment the sparse data and performance presented.

4.3.3 Segmentation with Sparse Range Data Only

The sparse laser range finder data from the Velodyne HDL-64E sensor is shown in Figure 4.5 with false coloring and hand-labeled truth segments to indicate the ideal segmentation of the scene. The scene contains four pedestrians, two walking next to each other and two near the bridge walls and another car on the road surface. The difficulty in segmenting this scene using the sparse range data alone is illustrated in Figures 4.6 and 4.7. The sparse range data is segmented using the RBNN approach from (Klasing et al., 2008). The RBNN performs well with a lower threshold (Figure 4.6) in identifying the pedestrians, but has difficulty creating a coherent cluster for the road surface because successive range scans are intersecting the road surface at increasing distance. To correct this problem, a potential solution is to increase the radial threshold, but the smallest threshold that clusters the road leads to nearly the entire range scan being segmented into a single cluster (Figure 4.7). The other problem is that the vehicle is in contact

with the road and this leads to difficulty in separating the vehicle from the road surface.

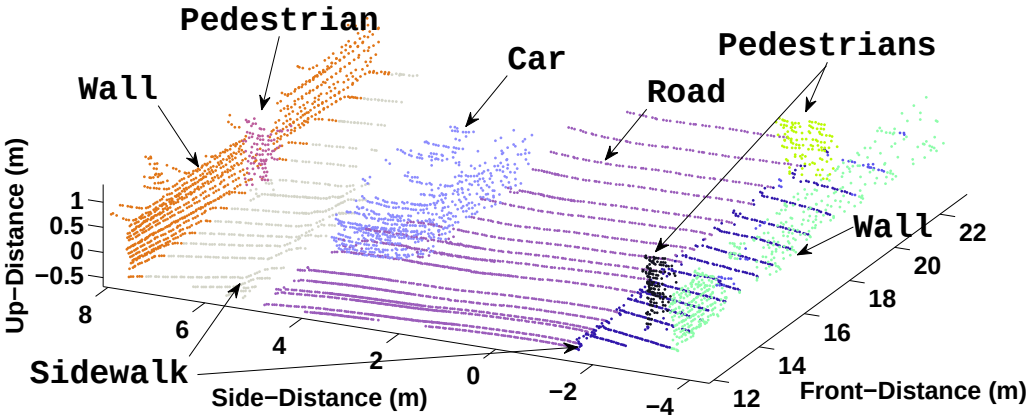

Figure 4.5: Labeled truth data of the sparse laser range finder data.

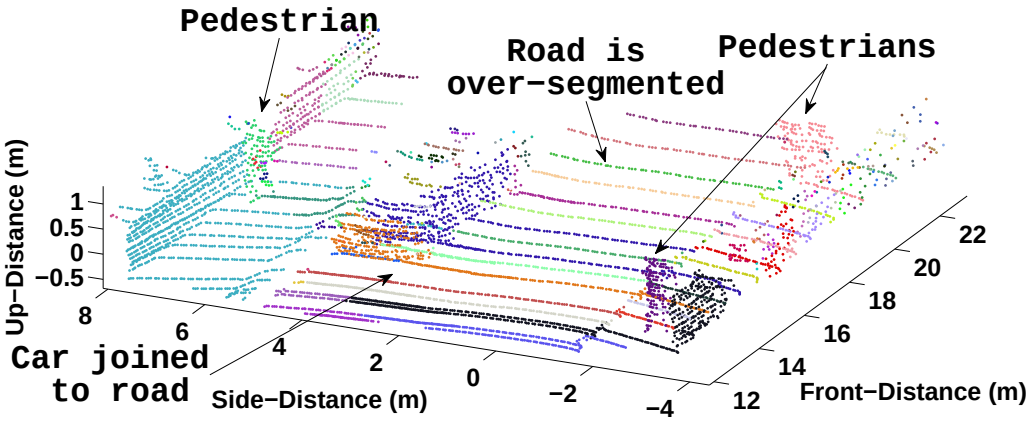

Figure 4.6: RBNN algorithm from (Klasing et al., 2008) shows over segmentation of the road surface due to increasing distance between scans intersecting the road.

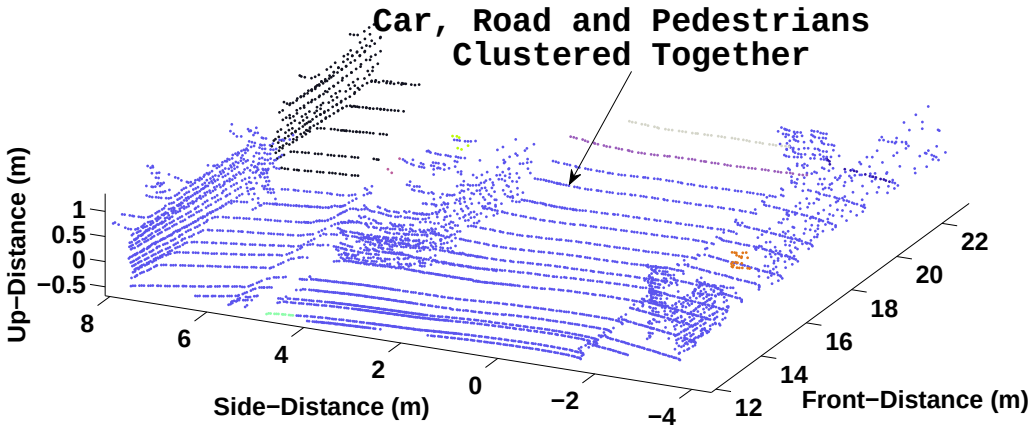

Figure 4.7: In order to cluster the road surface, the RBNN algorithm from (Klasing et al., 2008) creates nearly a single large cluster.

4.3.4 Segmentation with Sparse Range Data Augmented with Color and Surface Normal Estimates

To attempt to solve the problem of a large threshold causing nearly a single cluster and a low threshold on the range data causing over segmentation, a modification to the RBNN algorithm is presented. The color of each sparse point and the estimated surface normal at each point are used to construct the cost metric in (4.3) for use as the radial bound in the RBNN algorithm. The surface normals are estimated at each point using the PCA technique (Klasing et al., 2009) by joining the 8 nearest neighbors. Figure 4.8 shows the result from including pixel intensity and estimated surface normal into the segmentation of the sparse range data. The segmentation result shows the car is maintained as a set of separate clusters from the road, but that the road is not clustered into a coherent segment. Augmenting the sparse range data with the pixel intensity and surface normal estimates are not enough to achieve adequate segmentation performance.

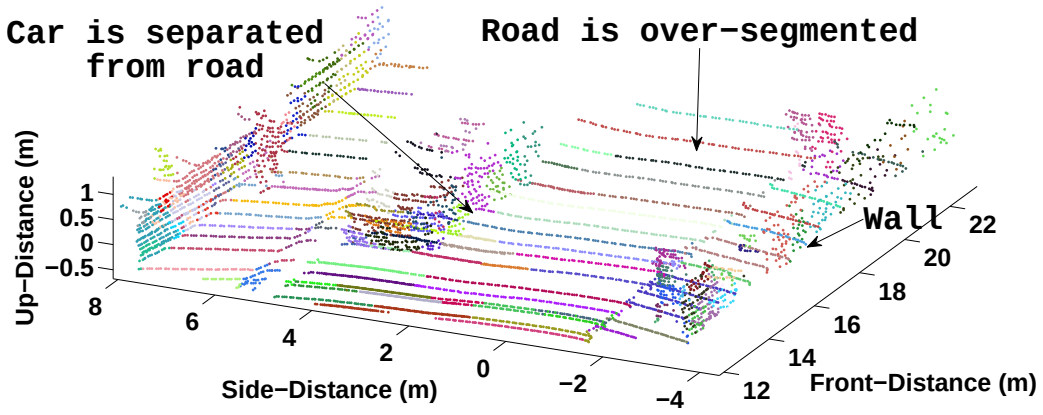

Figure 4.8: Including color and estimated surface normal with only the sparse range data still creates over-segmented range scans, motivating the use of the dense range map achieved through interpolation with the MRF.

4.3.5 Segmentation of the Textured Dense Range Map

Finally, the full textured dense point cloud performance is shown using the algorithm described in Section 4.2. The hand-labeled segments for the dense point cloud are shown in Figure 4.9. The final results of the segmentation of the dense range map are shown in Figure 4.10. The ability to estimate a 3D point at every pixel in the image provides contextual information for the algorithm to make decisions on what to segment. The road surface is a prime candidate to benefit from the algorithm due to the near continuous estimated points along the surface, thereby avoiding the need to handle increasingly larger spaces between successive line scans. The pedestrian along the left side of the image is joined with the wall because the interpolated range samples reduce the distance metric between the pedestrian samples and the wall.

Table 4.1 shows the computed performance calculated from hand-labeled truth

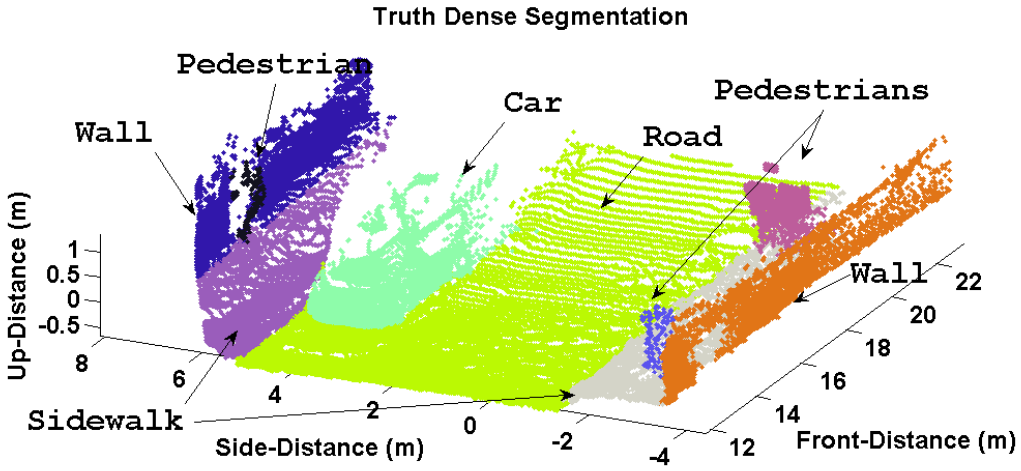

Figure 4.9: Labeled truth data of the dense laser range finder data shows continuity between the vehicle and the road, which provides a challenge for accurate segmentation.

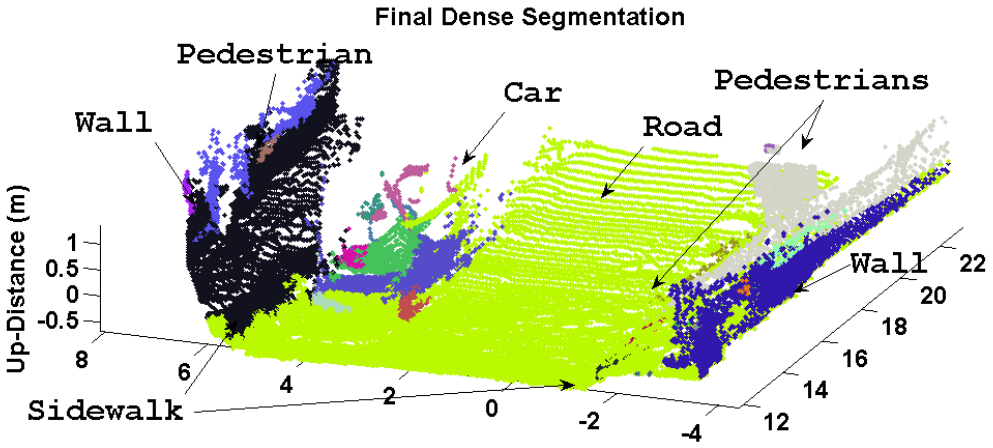

Figure 4.10: Final result of the newly proposed algorithm shows segmentation of the dense range data achieves separation of the car and a coherent road segment.

data over 13 frames around the one presented in detail. The ideal performance for each of the performance metrics is 0 for perfect segmentation and 1 for the worst segmentation. As expected, the RBNN algorithm with the low threshold is penalized in the OCE metric for over-segmentation and has a higher score for the GCE and LCE as well. The RBNN algorithm with the high threshold segments nearly the entire scene into a single cluster. This is also penalized in the OCE metric, but less penalized in the GCE and LCE metrics, leading to a lower score. The modified RBNN algorithm, which includes the color and estimated surface normal, has worse performance than the simpler RBNN algorithm because the scene is still heavily over-segmented. The benefit of segmenting the full dense 3D point cloud developed using Algorithm 1 is clear in the OCE, GCE, and LCE metrics, because the point cloud is not over-segmented nor clustered into a single segment. The largest cluster in all the frames analyzed is the road, therefore, when using the full dense 3D point cloud the accurate segmentation of the road leads to better performance in the OCE metric. The performance benefit when utilizing the dense map instead of the sparse map is apparent, because the sparse map is not adequately segmented even when the color and surface normal information is included.

Table 4.1: Segment performance for 13 total frames as compared with various metrics.

Algorithm	OCE	GCE	LCE
RBNN-Low	0.79	0.30	0.27
RBNN-High	0.80	0.14	0.09
Modified RBNN	0.89	0.16	0.14
Full Dense	0.53	0.06	0.07

4.3.6 Algorithm Run-Time

The algorithm run-time is dependent on the size of the camera image, because finding the MAP estimate to the MRF (4.1) requires the solution of n_{pix} simultaneous equations and the segmentation routine is linear in the number of pixels. The algorithm is run on a desktop computer with a 2.7GHz Intel®Core™i7 processor which enables different portions of the algorithm to be run as different threads each on separate cores. The run-time is summarized in Table 4.2 and shows that near real-time performance is achieved at full resolution with an update rate of 0.5 Hz; however, the half resolution image achieves a 2 Hz update rate. The 2 Hz update rate for this rich information is equivalent to the lane-finding algorithm used on the Cornell University DARPA Urban Challenge vehicle (Miller et al., 2008a), while providing much more operational usefulness.

Table 4.2: Dense point cloud segmentation algorithm run-time at full resolution and half resolution, along with sparse data segmentation routine run-time (msec).

Process	RBNN	Modified RBNN	Full Res. 337,920 pixels	Half Res. 84,480 pixels
Align Range Data	-	7	7	7
Solve MAP Estimate	-	-	385	100
Compute Normals	-	120	1170	320
Segment Point Cloud	116	51	291	86
Total (msec)	116	178	1852	513

4.4 Conclusion

An algorithm to segment textured dense 3D point clouds is presented and demonstrated in a complex urban environment. Textured dense point clouds are generated from interpolating sparse laser range finder data constrained by an aligned optical image. The dense point cloud is found using efficient matrix solver routines that enable near real-time performance. The segmentation is performed using an efficient and deterministic algorithm that utilizes the pixel intensity and 3D point geometry data. The algorithm is demonstrated in a complex urban environment which shows the successful ability to segment the road surface as a coherent cluster. The benefit of using the dense range data for interpolation is demonstrated by examining the segmentation performance with the sparse range data alone and augmenting the sparse range data with pixel intensity and surface normal estimates prior to segmentation. Finally, fast run-time of the algorithm provides operational usefulness for large-scale real-time autonomous vehicles.

4.5 Acknowledgments

The authors would like to thank Peter Moran and the Cornell University DARPA Urban Challenge Team for providing sensors, sensor interfaces, and data collection support. This work is supported under the Northrop Grumman Electronic Systems Scholars Program, ARO Grant #W911NF-09-1-0466 and NSF Cyber-Physical Systems Grant #CNS-0931686.

CHAPTER 5

CONSISTENT GENERALIZED DATA FUSION FOR MULTIPLE ROBOT OCCUPANCY GRID MAPPING

Abstract

This paper describes two information theoretic procedures for fusing multiple distributions with unknown correlation. In particular, the focus is finding fusion rules consistent with weighted exponential product (WEP) conservative fusion. A common approach to selecting the WEP fusion is to select the weight that equates the Kullback-Leibler divergence between the fused and constituent distributions; this is commonly called Chernoff Fusion. Chernoff Fusion does not account for any known discrepancy in the information used to form the constituent distributions. To account for this, the first information theoretic fusion approach developed here is Entropy Weighted Chernoff fusion; this fusion procedure biases the WEP fusion weight towards the distribution with the lowest entropy. This produces a fusion rule that accounts for the difference in exclusive information content in each distribution. An information loss for the WEP conservative fusion rule is also introduced and an approximation derived by computing the Kullback-Leibler divergence between the Naive Bayes and WEP fused distributions. The second information theoretic fusion approach is found by minimizing the approximation on information loss that occurs during WEP fusion. The resultant Minimum-Information-Loss Fusion rule generates the least conservative fused distribution in the family of WEP results. The two fusion rules are studied for the application of multi-robot occupancy grid mapping. Experimental results include the fusion of multiple occupancy grid maps over an optimally connected sensor network, demonstrating consistent map estimates.

5.1 Introduction

For sensor agents in a network to share information and perform distributed data fusion, it is desired to establish a scalable, flexible and robust network over which the robots can transmit and receive information. An ad-hoc and arbitrary connected network provides scalability for nodes to join and drop off the network, flexibility to allow nodes to join at any point and robustness to ensure multiple links or nodes must fail before the network becomes unconnected (Dekker and Colbert, 2004). To perform distributed data fusion over an arbitrary network topology requires conservative data fusion techniques to create consistent estimates that avoid rumor propagation (Chang et al., 2008). The techniques are conservative (or suboptimal), because they avoid double counting common information and discount exclusive information in the fused solution relative to the union of the information available in a centralized solution (Chang et al., 2010).

A common conservative fusion rule for Gaussian distributions with unknown correlation is Covariance Intersection (CI) (Julier and Uhlmann, 1997). For fusion of general probability distributions with unknown correlation, a conservative fusion technique, satisfying the rules of no double counting and discounting exclusive information, is the weighted exponential product (WEP) (Bailey et al., 2011). The WEP is a generalization of the CI method for non-Gaussian distributions (Hurley, 2002) and several approaches to determine the fusion weight to use for WEP have been proposed, leading to different fusion rules.

The simplest approach is to assign the WEP fusion weight to 0.5 and the resulting fusion rule is referred to as Bhattacharyya Fusion (Chang et al., 2008). If the WEP is motivated from decision theory, the weight applied during fusion is determined to minimize the error probability between two hypothesis corresponding

to the constituent distributions (Cover and Thomas, 1991). From this context, the WEP fusion weight is set to equate the Kullback-Leibler divergence metric between the resulting fused distribution and the each constituent distribution, this is called Chernoff Fusion (Cover and Thomas, 1991). The Chernoff Fusion nomenclature is from the fact that the resultant WEP fusion weight is equal to the Chernoff Information of the two constituent distributions. The WEP fusion approach is used here, but the weight does not correspond to the Chernoff Information, instead two novel fusion techniques are developed.

The first technique applies a factor, corresponding to the Entropy of each constituent distribution, to the desired equality of the Kullback-Leibler divergence between the resulting WEP fused distribution and the constituent distribution assumed in Chernoff Fusion. This biases the Chernoff Fusion resultant weight towards the distribution with lower entropy, thereby attempting to account for the additional exclusive information present in that distribution if a common prior is assumed. The resulting distribution from WEP fusion is shown to discount exclusive information and contains an information loss compared to the optimal fusion rule if the correlation between the two distributions were known. The second fusion technique is to minimize an approximate information loss metric that is later derived. Unlike Chernoff fusion, this approach determines the least conservative WEP fusion distribution. The WEP fusion is applicable to fusion of general distributions and the fusion of probabilistic environmental maps from autonomous robots across an arbitrary network is the focus here.

Mobile robots equipped with active sensors are effective at building probabilistic representations and maps of the environment in which they are operating (Thrun, 2002). As multiple robots operate in the same environment, it is advanta-

geous to share independently observed information with the other agents across a sensor network to form a more complete representation of the environment. This sharing accomplishes two goals. The first is to more completely cover the spatial environment when the received information contains a space of the environment previously unobserved by the receiving agent. The second is to reduce uncertainty about jointly observed and overlapping areas of the environment. The distributed mapping paradigm seeks to establish a data fusion procedure that builds a statistically consistent global map, using the information from each agent, without a centralized server, using only agent-to-agent communication and without global knowledge of the network topology.

In order to build a statistically consistent global map, the distributed data fusion procedure must avoid double counting common information between agents when performing fusion (Grime and Durrant-Whyte, 1994). One approach to avoid double counting information during fusion is to append pedigree information to all measurements (Ceruti et al., 2006). However, this approach requires redundant transmission of information to ensure it is received at all agents in the network and bandwidth constraints may preclude this action (Schoenberg and Campbell, 2009). Another approach is to constrain the global network topology to a tree and utilize the channel filter (Grime et al., 1992) to explicitly track common information between nodes and avoid double counting during fusion. The approach was extended to 2-tree and mixed 1&2-tree networks (Thompson and Durrant-Whyte, 2010) to increase network robustness. Unfortunately, these approaches violate the tenant that there is no global knowledge of the network topology, because the global network must be a tree topology. The Distributed Bellman-Ford algorithm (Awerbuch et al., 1994) can be invoked to discover the global network topology and artificially break links that violate the topological constraints. In practical

operation this network discovery algorithm must run in parallel to the data fusion process and requires reset procedures upon network reconfiguration.

To utilize generalized fusion across an ad-hoc network, the environmental representation should be a self-consistent probabilistic map. One possible metric map representation is the occupancy grid (Elfes, 1989), where the environment is partitioned into a lattice and each cell represents a probabilistic estimate of the occupancy. The lattice can be fixed or variable (Kraetzschmar et al., 2004) resolution, and either 2D or 3D (Fournier et al., 2007). A memory and throughput efficient 3D occupancy grid implementation (Wurm et al., 2010) is utilized in this work, because it provides the ability to create large scale, dense 3D maps with a reasonable computational and memory footprint. The multi-robot occupancy grid mapping problem has been evaluated previously, with focus on determining the unknown transformation between independent maps (Konolige et al., 2003). The problem can be solved with stochastic search to solve the optimization problem for the transformation between the two map to maximize the overlap (Carpin et al., 2005). The approach was later refined and error detection added (Birk and Carpin, 2006) and a new deterministic algorithm based on spectral decomposition of the map image (Carpin, 2008) was developed. However, none of these approaches dealt with the probabilistic fusion of maps after the best possible transformation was found. This work does consistent generalized probabilistic fusion using the WEP fusion rule and the transformations between maps are assumed known, but could be found via the above approaches.

The remainder of this paper introduces data fusion and generalized fusion in Section 5.2. A novel method for computing the WEP generalized fusion weight based on Entropy Weighted Chernoff fusion is presented in Section 5.2.1. A new

approximation on the information loss resulting from generalized fusion is derived in Section 5.2.2. A second novel WEP fusion weight computation approach based on minimizing the information loss resulting from generalized fusion is presented in Section 5.2.3. Section 5.4.1 introduces generalized fusion for occupancy grid maps and shows the fusion results for the new fusion techniques.

5.2 Generalized Data Fusion with Unknown Correlation

When the correlation between two probability distribution is unknown, the challenge in fusing the two distributions is a trade between maximizing resulting information content, yet maintaining consistency. In the information theoretic view of consistency, it is sufficient to avoid double counting information that is common between the two distribution. The resultant Bayes optimal closed form fusion rule that avoids double counting information is (Grime et al., 1992):

$$p(x|Z_i^K \bigcup Z_j^K) = \frac{1}{c}\frac{p(x|Z_i^K)p(x|Z_j^K)}{p(x|Z_i^K \bigcap Z_j^K)} \tag{5.1}$$

In practice, keeping track of the denominator in (5.1) is difficult for a distributed data fusion system on an arbitrary network without sharing pedigree about each distribution to fuse. The alternatives for consistent fusion include the weighted exponential product (WEP) conservative fusion rule (Cover and Thomas, 1991):

$$p_{\text{WEP}}(x|Z_i^K \bigcup Z_j^K) = \frac{1}{c}p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)} \tag{5.2}$$

$$0 \leq \omega \leq 1$$

which is generalizable to any probability distributions. However, the choice of WEP fusion weight ω is an open to trade.

If the weight ω in (5.2) is computed to equate the Kullback-Leibler divergence (D_{KL}) between the fused distribution $p_{\text{WEP}}(x|Z_i^K \bigcup Z_j^K)$ and the individual distributions (5.3):

$$\begin{aligned} &D_{\text{KL}}\left[p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)}||p(x|Z_i^K)\right] = \dots \\ &D_{\text{KL}}\left[p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)}||p(x|Z_j^K)\right] \end{aligned} \tag{5.3}$$

the resultant $\omega = \omega^*$ corresponds to the Chernoff Information (5.4) between $p(x|Z_i^K)$ and $p(x|Z_j^K)$ (Cover and Thomas, 1991):

$$w^* = C\left[p(x|Z_i^K), p(x|Z_j^K)\right] = -\arg\min_{\omega\in[0,1]} \log\left(\int p^{\omega}(x|Z_i^K)p^{1-\omega}(x|Z_j^K)dx\right) \tag{5.4}$$

and the WEP fusion rule (5.2) is called Chernoff Fusion (Julier, 2006).

An assumption as part of Chernoff Fusion is that the number of measurements used to estimate the probability distributions $p(x|Z_i^K)$ and $p(x|Z_j^K)$ is assumed unknown and equal (Hurley, 2001). If there is a measure of the total number of measurements used to estimate the density, this is used to bias the WEP fusion weight towards the distribution with more measurements. The focus of the sequel now turns to finding alternatives to Chernoff Fusion. The first fusion rule (Section 5.2.1) relaxes the assumption about equal measurements and applies a factor to (5.3) corresponding to the Entropy of the constituent distribution, thereby biasing the fusion weight towards the distribution with the lowest Entropy without explicitly counting measurements as suggested in (Hurley, 2001). The WEP fusion rule is shown to discount exclusive information (Rendas and Leita, 2010) from the constituent distributions, therefore, a second fusion approach (Section 5.2.3) is developed to shift the optimization criterion away from equating the Kullback-Leibler divergence between the resulting fused and constituent distributions (5.3)

to minimizing an approximation on the information lost as a result of fusion Section 5.2.3.

5.2.1 Entropy Weighted Chernoff Fusion

A commonly overlooked criterion for Chernoff fusion from a Bayesian decision theoretic viewpoint is that the number of measurements used to estimate the constituent distributions $p(x|Z_i^K)$ and $p(x|Z_j^K)$ for fusion are equal (Hurley, 2001). This is the case from the standpoint of hypothesis testing, but is not always true for arbitrary data fusion. Instead, Hurley suggests a means for finding ω by minimizing the total probability of error. This can be accomplished by selecting ω such that:

$$\begin{aligned} c_i \cdot D_{\mathrm{KL}}\left[p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)} || p(x|Z_i^K)\right] = \ldots \\ c_j \cdot D_{\mathrm{KL}}\left[p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)} | p(x|Z_j^K)\right] \end{aligned} \tag{5.5}$$

where c_i and c_j are the number of measurements used to estimate the probability distributions $p(x|Z_i^K)$ and $p(x|Z_j^K)$ respectively. In this work, there is a desire to avoid explicitly counting the number of measurements used to estimate probability distributions to fuse. Therefore, the c_i and c_j are estimated to be the reciprocal of the entropy of the initial distributions:

$$c_i = \frac{1}{H[p(x|Z_i^K)]} \tag{5.6}$$

$$c_j = \frac{1}{H[p(x|Z_j^K)]} \tag{5.7}$$

where $H[p(x)]$ is the entropy of the corresponding distribution $p(x)$. Therefore, the Entropy Weighted Chernoff Fusion solution uses a weight $\omega^* \in [0, 1]$ that

equates the entropy weighted Kullback-Leibler Divergence between the WEP and the original distributions:

$$\frac{1}{H[p(x|Z_i^K)]} \cdot D_{\text{KL}}\left[p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)}||p(x|Z_i^K)\right] = \dots$$
$$\frac{1}{H[p(x|Z_j^K)]} \cdot D_{\text{KL}}\left[p(x|Z_i^K)^{\omega} \cdot p(x|Z_j^K)^{(1-\omega)}||p(x|Z_j^K)\right] \tag{5.8}$$

This results in a bias towards the distribution with the lowest entropy along the chord that connects the two distributions, according to the Kullback-Leibler divergence metric.

5.2.2 Information Loss and Approximation

The fused density resulting from WEP fusion (5.2) discounts exclusive information (Rendas and Leita, 2010), while maintaining common information. The WEP method discounting of exclusive information ensures there is no double counting of common information from the two distributions during the fusion process, when the correlation between the distributions is unknown. Therefore, there is an information loss between the optimal fusion (5.1) and the WEP fusion (5.2). This information loss is defined here using the Kullback-Leibler divergence between the optimal (5.1) and the WEP solutions (5.2):

$$\mathcal{I}_{\text{LOSS}} \triangleq D_{\text{KL}}\left[p_{\text{OPTIMAL}}(x|Z_i^K \bigcup Z_j^K)||p_{\text{WEP}}(x|Z_i^K \bigcup Z_j^K)\right] \tag{5.9}$$
$$\tag{5.10}$$

Unfortunately, without knowing the optimally fused distribution, it is not possible to compute the information loss metric (5.9). However, if the correlation between

the two distributions to fuse is known to be zero, the optimal fusion rule simplifies to the Naive Bayes (NB) fusion (5.11):

$$p_{\text{NB}}(x|Z_i^K \bigcup Z_j^K) = \frac{1}{c} p(x|Z_i^K) p(x|Z_j^K) \tag{5.11}$$

In the case of an unknown correlation between the two distributions to fuse, the discounting of exclusive information resultant from WEP fusion (5.2) leads to information loss if the true correlation between the distribution is zero. Therefore, an approximation to the information loss is the Kullback-Leibler divergence between the Naive Bayes fusion and WEP fusion:

$$\mathcal{I}_{\text{LOSS}} \approx \bar{\mathcal{I}}_{\text{LOSS}}(\omega) = D_{\text{KL}} \left[p_{\text{NB}}(x|Z_i \bigcup Z_j) || p_{\text{WEP}}(x|Z_i \bigcup Z_j) \right] \tag{5.12}$$

It is now possible to compute this information loss approximation and to derive a fusion rule based on minimizing the amount of information loss compared to the Naive Bayes fusion.

5.2.3 Minimum Information Loss Weight Fusion

Given that the information loss is approximated by the Kullback-Leibler divergence between the Naive Bayes solution and the WEP fusion, the choice of fusion weight $\omega^* \in [0, 1]$ is selected to minimize the information loss (Minimum-Information-Loss Fusion).

$$\omega^* = \underset{\omega \in [0,1]}{\arg\min}\, \bar{\mathcal{I}}_{\text{LOSS}}(\omega) \tag{5.13}$$

which is expanded to:

$$= \arg\min_{\omega\in[0,1]} D_{\text{KL}}\left[P_{\text{NB}}(x|Z_i,Z_j)||P_{\text{WEP}}(x|Z_i,Z_j)\right] \tag{5.14}$$

$$= \arg\min_{\omega\in[0,1]} \int \frac{1}{c_{\text{NB}}} p(x|Z_i)p(x|Z_j)\log\left[\frac{c_{\text{WEP}}p(x|Z_i)p(x|Z_j)}{c_{\text{NB}}p(x|Z_i)^{\omega}p(x|Z_j)^{1-\omega}}\right]dx \tag{5.15}$$

$$= \arg\min_{\omega\in[0,1]} \int \frac{1}{c_{\text{NB}}} p(x|Z_i)p(x|Z_j)\log\left[\frac{c_{\text{WEP}}}{c_{\text{NB}}}p(x|Z_i)^{(1-\omega)}p(x|Z_j)^{\omega}\right]dx \tag{5.16}$$

The advantage of the Minimum-Information-Loss Fusion (5.13) scheme is that while the WEP fusion rule (5.2) is still conservative and consistent, the weight is selected to minimize the possible information loss (5.9) that results should the two distributions be truly uncorrelated. In practice, Minimum-Information-Loss Fusion drives the solution towards Naive Bayes fusion in the case when the distributions to fuse are significantly different (in a Kullback-Leibler sense), thereby indicating the two do not share a significant amount of common information. The Minimum-Information-Loss Fusion scheme provides an automatic method for trading between the Naive Bayes and conservative fusion rules, without resorting to a heuristic decision.

5.3 Consistent Generalized Fusion

To evaluate the impact of the newly developed generalized fusion rules, an in-depth evaluation of the rules applied to Bernoulli distributions is performed. The Bernoulli distribution provides a simple distribution to compute the fusion rules in closed and show the general trend differences that occur between the different fusion rules. The analysis includes examination of Chernoff Fusion (5.4), Entropy Weighted Chernoff Fusion (5.8) and Minimum-Information-Loss Fusion (5.13).

5.3.1 Generalized Fusion of Bernoulli Distributions

This section analyzes the WEP fusion weight, resulting fused distributions and approximate information loss across all possibilities of two Bernoulli distributions. Applying the Weighted Exponential Product (WEP) generalized fusion rule (5.2) to two Bernoulli distributions (5.27) with unknown correlation results in a closed form solution:

$$p(x|Z_i^K) = p_i \tag{5.17}$$

$$p(x|Z_j^K) = p_j \tag{5.18}$$

$$p_{\text{WEP}}(x|Z_i^K \bigcup Z_j^K) = p_\omega = \frac{p_i^\omega p_j^{1-\omega}}{p_i^\omega p_j^{1-\omega} + (1-p_i)^\omega (1-p_j)^{1-\omega}} \tag{5.19}$$

To numerically evaluate the outcome of consistent generalized fusion over a Bernoulli distribution, two Bernoulli probabilities p_i and p_j are discretized in steps of 0.01 from $[0, 1]$. For each step of each probability, the resulting WEP fusion weight is computed according to the desired fusion rule. The resulting WEP fused density (probability) (5.19) is computed for all combinations of p_i and p_j. The approximate information loss $\bar{\mathcal{I}}_{\text{Loss}}(\omega)$ (5.12) is computed based on the fusion weight and fused density.

In the case of Chernoff Fusion (CF) (5.4), the optimal weight fusing of two Bernoulli distributions is solved in closed form:

$$\omega_{\text{CF}}^* = \frac{\log \frac{[\log(1-p_i) - \log(1-p_j)]}{[\log(p_i) - \log(p_j)]} - \log p_j + \log(1-p_j)}{[\log p_i - \log p_j - \log(1-p_i) + \log(1-p_j)]} \tag{5.20}$$

Figure 5.1 shows the resultant Chernoff fusion weight (a), fusion result p_ω (b) and approximate information loss (c). The Chernoff fusion weight (Figure 5.1 a) is close to 0.5 for almost all combinations of p_i and p_j. The exceptions are

when $p_i \approx 0.5$ and $p_j \approx 0, 1$ and the weight moves away from 0.5 to select the distribution with lowest entropy, because the Kullback-Leibler equality constraint requires more weight on the distribution with lower entropy. The resulting Chernoff Fusion probability (Figure 5.1 b) shows a smooth variation towards a fusion result that selects the distribution with more extreme (towards 0 or 1) probability. This is evidence from the large area that the fusion result $p_\omega \approx 1.0$ or $p_\omega \approx 0.0$ that occurs regardless of the probability of the other distribution. Finally, as $p_i = 1 - p_j$ the resulting fused density is $p_\omega = 0.5$ demonstrating how the fusion rule selects the fusion distribution equidistant from each constituent distribution, which is necessarily $p_\omega = 0.5$ for Bernoulli distributions. The approximate information loss for the Chernoff fusion (Figure 5.1 c) shows the potential problem with Chernoff fusion. The approximate information loss $\bar{\mathcal{I}}_{\text{LOSS}} \approx 0.0$ when $p_i \approx 1 - p_j$, which shows the Chernoff Fusion result tends towards the Naive Bayes Fusion in that case. However, when $p_i >> p_j$ or $p_j >> p_i$ the loss increases. This loss occurs, because the Chernoff solution discounts the maximum amount of exclusive information in each constituent distribution to find the fused distribution equidistant (in the Kullback-Leibler sense) from the two distributions to fuse. The Entropy-Weighted Chernoff Fusion and Minimum-Information-Loss Fusion will address this shortcoming in Chernoff fusion.

For the Entropy Weighted Chernoff Fusion (EWCF) (5.8), the weight ω^* equating the entropy weighted Kullback-Leibler divergence cannot be solved for in closed form, therefore, numerical equation-solving is necessary for the following function:

$$\begin{aligned} &\frac{1}{H[p_i]}\left[p_\omega \log \frac{p_\omega}{p_i} + (1-p_\omega)\log\frac{(1-p_\omega)}{(1-p_i)}\right] - \dots \\ &\frac{1}{H[p_j]}\left[p_\omega \log \frac{p_\omega}{p_j} + (1-p_\omega)\log\frac{(1-p_\omega)}{(1-p_j)}\right] = 0 \end{aligned} \tag{5.21}$$

where p_ω is found from (5.19) and the entropy of a Bernoulli distribution is $H[p] =$

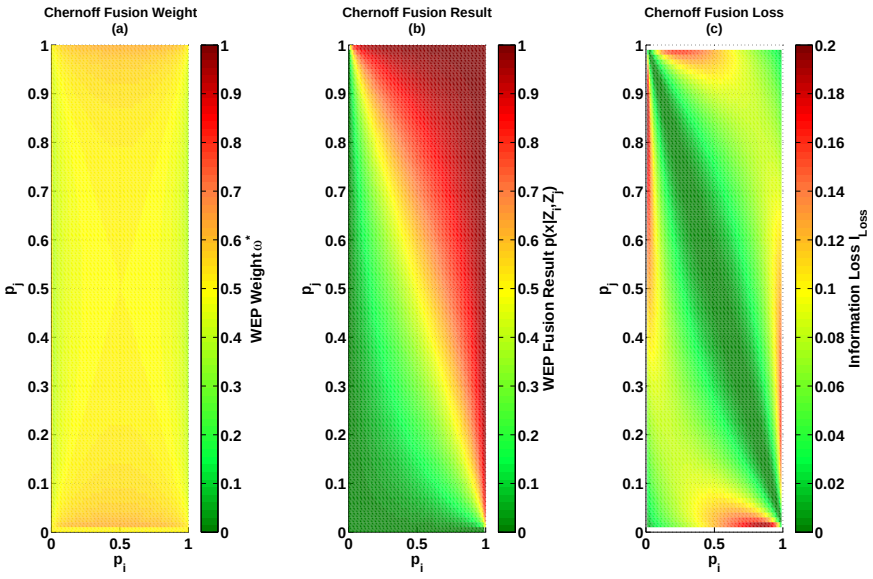

Figure 5.1: Chernoff Fusion over two Bernoulli distributions with the optimal weight (a), fusion result (b) and approximate information loss (c).

$-p \log p - (1-p) \log(1-p)$. When the entropy of the two distributions is equal, the Entropy Weighted Chernoff Fusion collapses to Chernoff Fusion and the weight is solved for in closed form according to (5.20). The entropy of two Bernoulli distributions is equal $H[p_i] = H[p_j]$, when $p_i = p_j$ or $(1 - p_i) = p_j$, because the entropy is symmetric about $p = 0.5$.

Figure 5.2 shows the resultant Entropy Weighted Chernoff fusion weight (a), fusion result p_ω (b) and approximate information loss (c). The results of the EWCF weight (Figure 5.2 a) are similar to the CF (Figure 5.1 a), but have a sharper transition away from $\omega^* = 0.5$ when the two distributions are significantly different from one another. Although difficult to visualize, for a fixed p_i, the weight as a function of $p_j \in [0, 1]$ (and vice versa) is asymmetric about $p_j = 0.5$, due to the bias introduced by the entropy weighting factor. In addition, as $p_i \approx 0, 1$ or $p_j \approx 0, 1$ the weight more quickly (compared to CF) tends to select the distribution at the extreme through the bias introduced by the entropy. The resulting EWCF

probability (Figure 5.2 b) again shows a similar trend to the CF. The difference is that the transition from the fusion result being at the extreme $p_\omega \approx 0, 1$ is a sharper transition than in Chernoff Fusion. In addition, as $p_i >> p_j$ or $p_j >> p_i$ the fusion result tends towards maximum uncertainty $p_\omega \approx 0.5$ less quickly compared to CF. The impact of the bias towards the distribution with lowest entropy is visible in the approximate information loss of the EWCF (Figure 5.2 c) that shows a reduced information loss compared to CF. The areas of higher information loss in CF (Figure 5.1 c) occur when $p_i >> p_j$ or $p_j >> p_i$ and the solution maximally discounts exclusive information. The EWCF accounts for the possible difference in exclusive information and computes a fused density that is closer to Naive Bayes and the resulting information loss is reduced. The EWCF has the same information loss when $p_i \approx p_j$ or $p_i \approx (1 - p_j)$, because EWCF reduces to CF in these cases. This includes the tendency towards zero information loss when $p_i \approx (1 - p_j)$.

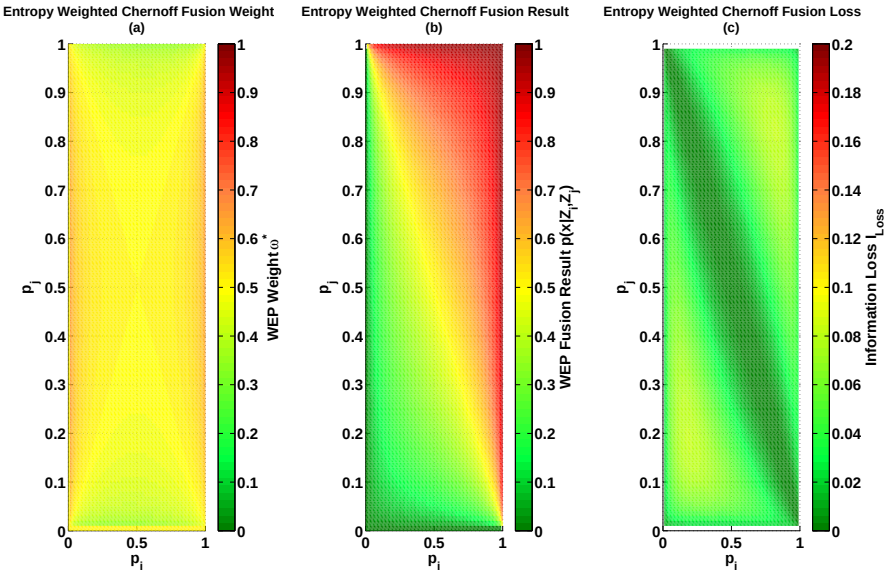

Figure 5.2: Entropy Weighted Chernoff Fusion over two Bernoulli distributions with the optimal weight (a), fusion result (b) and approximate information loss (c).

In the case of the Minimum-Information-Loss Fusion (M-LF), the optimal

weight is again solved for in closed form:

$$\omega^*_{\text{M-L}} = \frac{\log p_j - \log(1-p_j) - \log\left[\frac{p_i p_j}{(p_i-1)(p_j-1)}\right]}{\log p_j - \log(1-p_j) - \log p_i + \log(1-p_i)} \tag{5.22}$$

Figure 5.3 shows the resultant Minimum-Information-Loss WEP fusion weight (a), fusion result p_ω (b) and approximate information loss (c). The Minimum-Information-Loss Fusion exhibit markedly different tendencies in the fusion weight (Figure 5.3 a) compared to the CF and EWCF. In particular, there is a boundary at $p_i = 0.5$ or $p_j = 0.5$ and if p_i and p_j are both less than or greater than 0.5. In the case when both $p_i < 0.5$ and $p_j < 0.5$ or $p_i > 0.5$ and $p_j > 0.5$ the optimal Minimum-Information-Loss weight (5.22) violates the constraint $\omega \in [0, 1]$ and therefore equals extreme $\omega = 0, 1$ in an effort to minimize the approximate information loss. When $p_i < 0.5$ and $p_j > 0.5$ or $p_i > 0.5$ and $p_j < 0.5$, the M-LF weight is mirrored symmetric about $p_i = 1 - p_j$ and the weight is found to minimize the approximate Information Loss to zero. The M-LF result (Figure 5.3 b) shows similar trends to CF and EWCF, but in the cases where $p_i < 0.5$ and $p_j < 0.5$ or $p_i > 0.5$ and $p_j > 0.5$, the M-LF weight goes to $\omega = 0, 1$ and therefore just selects one distribution or the other in the fusion result, instead of using a combination of the two distributions. This discounts all of the exclusive information in one distribution in favor of keeping all the exclusive (and common) information in the other distribution. The resulting approximate information loss for M-LF (Figure 5.3 c) shows the benefit of the M-LF where the information loss goes to zero where $p_i < 0.5$ and $p_j > 0.5$ or $p_i > 0.5$ and $p_j < 0.5$. This shows the M-LF result is equal to the Naive Bayes solution, because the Kullback-Leibler divergence defining the information loss (5.12) is zero if and only if the WEP fused and Naive Bayes fused distributions are equal. This shows the M-LF rule provides a means to avoid discounting exclusive information for fusion of two Bernoulli distributions.

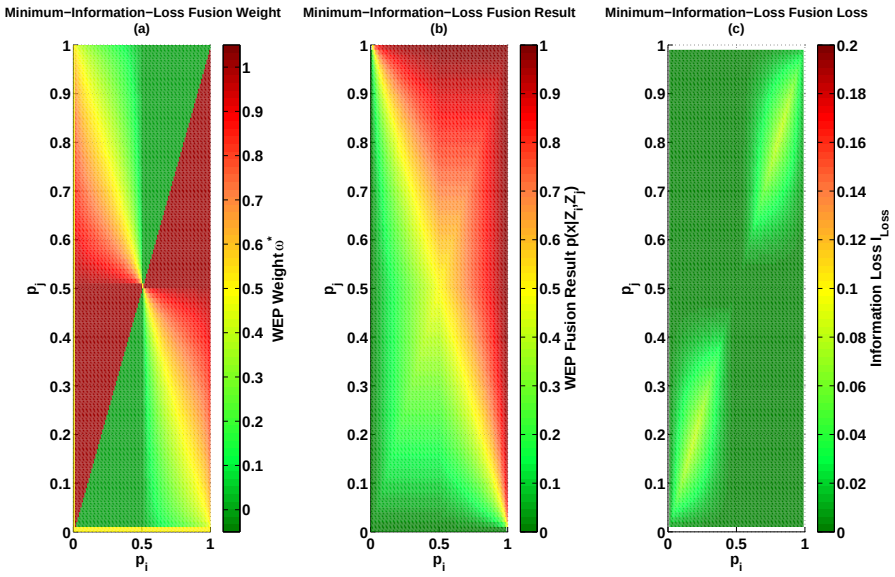

Figure 5.3: Minimum-Information-Loss WEP Fusion over two Bernoulli distributions with the optimal weight (a), fusion result (b) and approximate information loss (c).

Examining Figures 5.1-5.3 (c) shows qualitative differences in the fusion rules and the resulting information loss (5.12). To compute the effectiveness of each fusion approach, the empirical cumulative distribution function (CDF) is computed for the approximate information loss $\bar{\mathcal{I}}_{\text{LOSS}}$ for each of the three fusion rules: Chernoff Fusion (5.20), Entropy Weighted Chernoff Fusion (5.21) and Minimum-Information-Loss Fusion (5.22). In addition, the empirical CDF of $\bar{\mathcal{I}}_{\text{LOSS}}$ for Bhattacharyya fusion is computed for reference, because of the simple fusion rule $\omega^* = 0.5$ for all possible combinations of p_i and p_j.

Figure 5.4 plots the empirical CDF for $\bar{\mathcal{I}}_{\text{LOSS}}$ from (5.12) for each of the different fusion rules. The plot shows the probability of occurrence for each amount of $\bar{\mathcal{I}}_{\text{LOSS}}$. The Bhattacharyya Fusion, Entropy Weighted Chernoff Fusion, and Minimum-Information-Loss Fusion all have the same largest approximate information loss $\bar{\mathcal{I}}_{\text{LOSS}} = 0.08$, while Chernoff Fusion has a larger maximum $\bar{\mathcal{I}}_{\text{LOSS}} = 0.19$. The CDF

curves show the Minimum-Information-Loss Fusion approach drives the approximate information loss to zero for 50% of the possible combinations of p_i and p_j, whereas the other fusion rules have a longer tail probability for larger $\bar{\mathcal{I}}_{\text{Loss}}$.

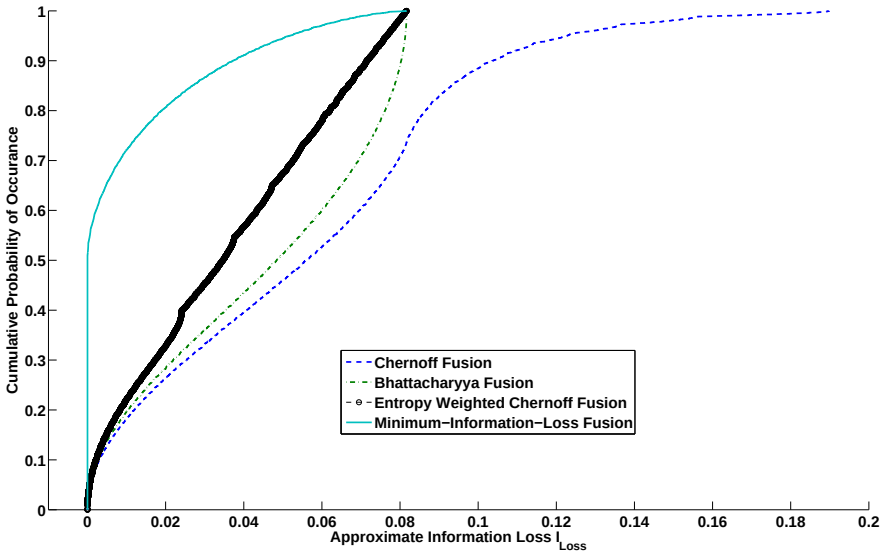

Figure 5.4: The CDF for the approximate information loss over Chernoff, Bhattacharyya, Entropy Weighted Chernoff and Minimum-Information-Loss WEP fusion.

5.3.2 Generalized Fusion of Arbitrary Distributions

The Bernoulli distribution was studied in-depth in Section 5.3.1, because of the simplicity of the Bernoulli distribution, the ability to solve in closed form for the Chernoff Fusion and Minimum-Information-Loss Fusion results, and the ability to visually and quantitatively examine trends in the different fusion rules over all possible Bernoulli probabilities. The WEP fusion (5.2) is general and can be used with any probability distributions $p(x|Z_i^K)$ and $p(x|Z_j^K)$. For arbitrary probability distributions there are three challenges associated with Chernoff Fusion, Entropy

Weighted Chernoff Fusion and Minimum-Information-Loss Fusion. The first is to compute the resultant WEP fusion (5.2) distribution. Not all probability distributions have a proper representation when raised to an arbitrary power and the product is not necessarily a proper distribution in the same family. The second challenge is computation of the fusion weight. In the case of Chernoff Fusion (5.4), and Minimum-Information-Loss Fusion (5.13) a minimization must be performed that is not always able to be solved in closed form. For Entropy-Weighted Chernoff Fusion solving directly for the weight in the equality (5.5) does not have a closed form. Therefore, numerical equation-solving of the objective functions is necessary. Finally, the third challenge is that simply evaluating the objective functions associated with each fusion approach is not always possible in closed form. Therefore, sampling methods might be necessary to evaluate and then solve or minimize the objective function via numerical techniques.

For example, if the constituent distributions to fuse are Gaussian, the resulting WEP fusion (5.2) is a Gaussian, where the mean and variance are found via the formulas given for Covariance Intersection (Julier and Uhlmann, 1997). However, solving for the weight to use in Covariance Intersection requires numerical calculation of a cost metric such as the determinant or trace of the fused covariance. The computation of the Chernoff Fusion (5.4), Entropy Weighted Chernoff Fusion (5.5) and Minimum-Information-Loss Fusion (5.13) weight for Gaussian distributions is not closed form and requires numerical optimization. However, in each case, the resultant optimization function can be evaluated in closed form. If the constituent distributions are Gaussian mixtures, the WEP fusion is not closed form, although approximations have been proposed (Julier, 2006). In addition, sampling techniques are necessary to evaluate the objective functions used in the different fusion rules that cannot be written in closed form. Future work is necessary and has

already begun toward this goal.

5.3.3 Conservativeness of WEP Fusion

WEP fusion is a conservative fusion rule that avoids double-counting common information contained in the two distributions to fuse. Consider the case of a fusion of two normal distributions $p(x|Z_i) \sim \mathcal{N}(\mu_i, P_i)$ and $p(x|Z_j) \sim \mathcal{N}(\mu_j, P_j)$. In the case of WEP fusion (5.2), the resultant fused Gaussian distribution has mean and covariance (Hurley, 2002):

$$\mu_{\text{WEP}} = P_{\text{WEP}} \left[\omega P_i^{-1} \mu_i + (1-\omega) P_j^{-1} \mu_j \right] \tag{5.23}$$

$$P_{\text{WEP}} = \left[\omega P_i^{-1} + (1-\omega) P_j^{-1} \right]^{-1} \tag{5.24}$$

If the common information between the two distributions is the empty set, such that $Z_i \bigcup Z_j = \emptyset$, then the optimal fusion rule (5.1) collapses to the Naive Bayes Fusion (5.11) and the resultant fused Gaussian distribution has mean and covariance (Chang et al., 2008):

$$\mu_{\text{NB}} = P_{\text{NB}} \left[P_i^{-1} \mu_i + P_j^{-1} \mu_j \right] \tag{5.25}$$

$$P_{\text{NB}} = \left[P_i^{-1} + P_j^{-1} \right]^{-1} \tag{5.26}$$

To illustrate the conservativeness of WEP fusion, consider a case where the two distributions are identical standard normal distributions $\mathcal{N}(0,1)$. In the case of WEP fusion, regardless of the choice of fusion weight ω, the resulting fused distribution is an identical standard normal: $p_{\text{WEP}}(x|Z_i \bigcup Z_j) \sim \mathcal{N}(0,1)$. In the case where there is no exclusive information, such that $Z_i = Z_j$, then the WEP fusion result is the correct optimal fusion. However, in the case that the two distributions do not share any common information, the Naive Bayes fusion produces a fused

distribution: $p_{\text{NB}}(x|Z_i \bigcup Z_j) \sim \mathcal{N}(0, \frac{1}{\sqrt{2}})$. The resulting distributions are shown in Figure 5.5 and graphically demonstrate the conservativeness of WEP fusion, because of the increase variance in the WEP fusion compared to the Naive Bayes fusion that is correct when there is no common information. This example provides

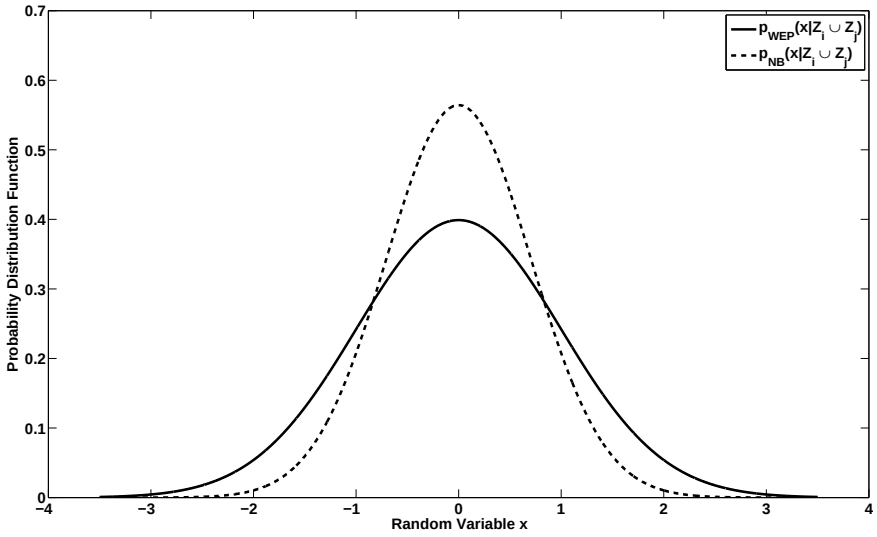

Figure 5.5: WEP and Naive Bayes Fusion results from fusion of two identical standard normal distributions.

a cautionary note about the conservativeness of WEP fusion, because while the mean remains the same in both cases, the standard deviation of the WEP fused estimate is conservative by a factor of $\sqrt{2}$.

5.4 Mulit-Robot Occupancy Grid Map Fusion

The generalized fusion of the Bernoulli distribution is directly applicable to fusion of multi-robot occupancy grid maps. The probability of a voxel of space being occupied is represented as a Bernoulli probability and each voxel is assumed independent. As a result, the fusion of occupancy grid maps from multiple robots can

be accomplished by sequentially performing fusion across each of the grid cells in the map.

The occupancy grid concept is reviewed in Section 5.4.1. Next in Section 5.4.2 a laboratory experiment demonstrates the fusion of occupancy grid maps based on the different fusion rules derived, including Chernoff Fusion, Entropy Weighted Chernoff Fusion and Minimum-Information-Loss fusion and compares the resulting approximate information loss as a function of map location for the different techniques.

5.4.1 Generalized Fusion of Occupancy Grid Maps

The occupancy grid mapping paradigm establishes a consistent metric map of the environment using noisy measurements from sensors. The occupancy grid represents each cell in the environment via a probability density of a binary variable: the cell is occupied or empty. The posterior probability of the cell c occupancy using all data Z^k up to time K is assumed to be Bernoulli distributed:

$$p(c|Z^K) = \begin{cases} q = 1 - p & \text{empty} \\ p & \text{occupied} \\ 0 & \text{otherwise} \end{cases} \tag{5.27}$$

$$p \in [0, 1]$$

The typical occupancy grid map implementation stores the log-odds of the posterior probability of the cell c occupancy, which can be updated efficiently

through recursive operations(Thrun, 2002):

$$c^K_{\text{log-odds}} = \log\left[\frac{p(c|Z^K)}{1-p(c|Z^K)}\right] \tag{5.28}$$

$$= \log\left[\frac{p(c|Z_k)}{1-p(c|Z_k)}\right] + \log\left[\frac{p(c|Z^{K-1})}{1-p(c|Z^{K-1})}\right] + \log\left[\frac{1-p(c)}{p(c)}\right] \tag{5.29}$$

$$= c_{\text{log-odds}_k} + c^{K-1}_{\text{log-odds}} \tag{5.30}$$

where (5.30) is the recursive Bayes update of the log-odds of the posterior probability of the cell c occupancy that arrives from setting the prior occupancy probability $p(c) = 0.5$. The map is updated from the inverse sensor model $p(c|Z_k)$ (Wurm et al., 2010) and the cell occupancy probability for the Bernoulli distribution in (5.27) is found via:

$$p(c|Z^K) = \frac{\exp c^K_{\text{log-odds}}}{\exp c^K_{\text{log-odds}} + 1} \tag{5.31}$$

An important assumption in occupancy grid maps that is used here is that the individual cells are independent of one another. This assumption enables generalization fusion of the occupancy grid maps to collapse to the fusion on individual grid cells occupying the same space.

5.4.2 Multi-Robot Laboratory Experiment

The data collection was performed in Cornell University's Autonomous Systems Laboratory (ASL) using the Pioneer P3-DX differential drive mobile robot from Mobile Robots Inc. shown in Figure 5.6. The primary sensor for the occupancy grid mapping is the Hokuyo URG-04X laser scanner which features a 240° field-of-view and angular resolution of 0.36° and maximum range of ≈ 5 m. The laser is pitched downward 45° and scans along the ground as the robot moves forward in a push-broom fashion.

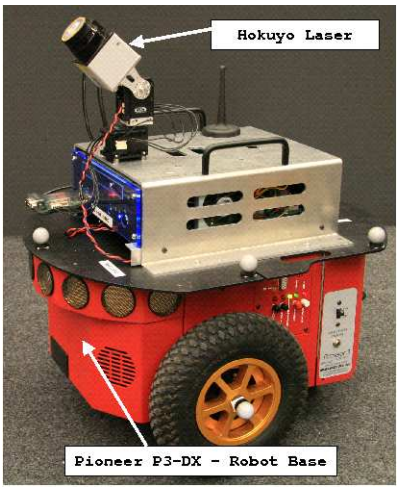

Figure 5.6: Mobile robot used in Cornell University's ASL equipped with Hokuyo laser scanner and on-board computer. The robot is used to explore and generate occupancy grid maps for distributed fusion in the laboratory experiment.

The test environment is 15×8 meters and is instrumented with a Vicon MX+ precision tracking system that determines position and attitude of 3D objects (notice tracking markers on the mobile robot in Figure 5.6) in the test environment and is used to localize the robots. In certain areas of the test environment, precision tracking is not available and integrated odometry is used to localize the robots with a corresponding increase in pose uncertainty. The features in the environment consist of boxes between 10 and 25 cm tall that are meant to simulate traffic cones or other similarly sized obstacles for a full-size traffic vehicle.

Eight robots are run in different paths around the environment for a 120 second data collection (Figure 5.8). The robots are run sequentially to avoid sensing another robot (dynamic objects) during the map construction. The paths of the robots are shown in Figure 5.8.

The robots each construct a 3D occupancy grid map using the Octomap (Wurm et al., 2010) implementation. The occupancy grid resolution is 0.05 m. To establish

Figure 5.7: Laboratory environment for the occupancy grid mapping experiment is 15×8 meters and contains boxes of different sizes.

a baseline occupancy grid map, the data from all the mobile robots is processed to make the *centralized* solution that is equivalent to all agents sending each laser scan and position report to a central server for processing. The resulting centralized map is shown in Figure 5.9. The map is rendered displaying only the voxels that are occupied or have a $p(c|Z^K) > 0.8$, the empty and unknown voxels are not shown and the voxels are falsely colored according to height; a 0.05 m grid is shown in gray for reference. The green voxels indicate areas of the maps that are safe for the robot to traverse. The line of small skinny boxes to the left of the environment are clearly visible along with the larger boxes towards the right of the environment and the outer walls. There are gaps in the center of the map where no robot explored. The centralized solution establishes a baseline map for use in the distributed occupancy grid mapping approaches.

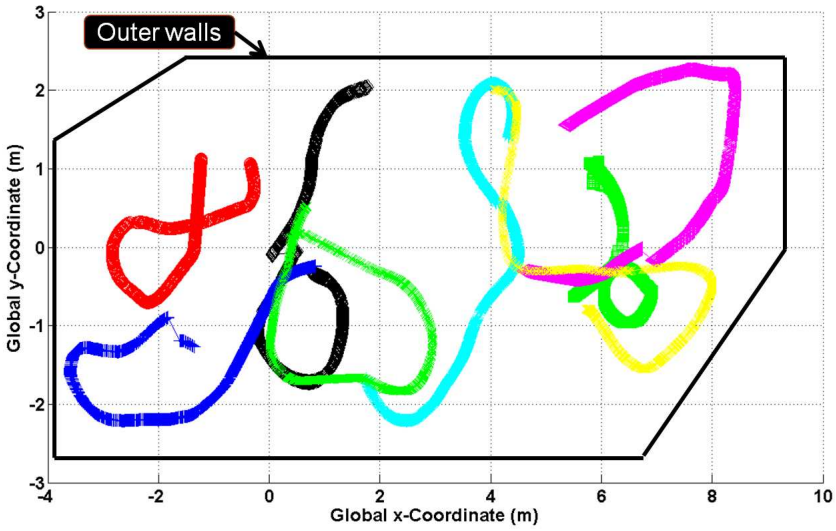

Figure 5.8: The overhead paths of the eight mobile robots exploring the environment for distributed occupancy grid mapping experiment. The data collection interval is 120 seconds and different robots explore different regions of the environment with some overlap to demonstrate the benefit of the generalized fusion rules to distributed mapping.

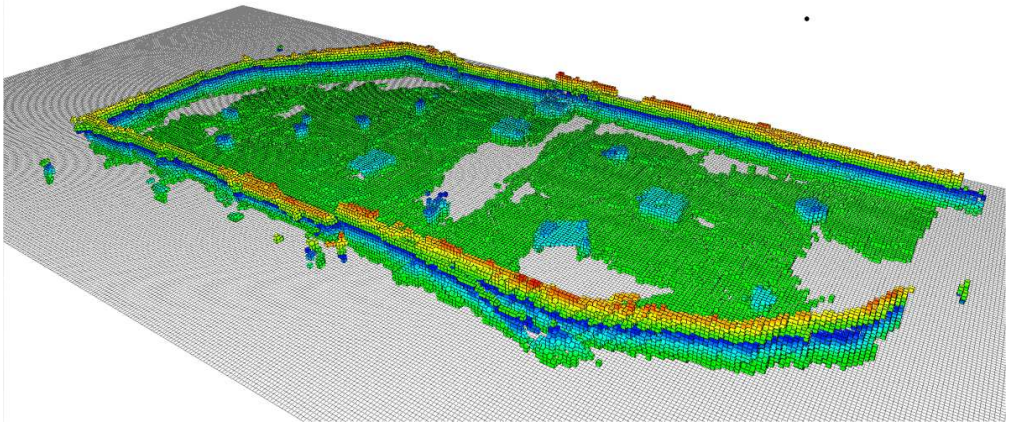

Figure 5.9: 3D centralized occupancy grid map displaying the occupied voxels with false coloring based on height.

5.4.3 Distributed Occupancy Grid Fusion on Optimally Connected Network

To evaluate the application of the distributed generalized fusion rules to occupancy grid mapping, the mobile robots are connected in a network. The mobile robots are the sensor nodes that are connected to form a sensor network. The network is connected according to the topology with maximum robustness (Dekker and Colbert, 2004). The topology is symmetric, has equal node connectivity and link connectivity that is equal to the minimum degree, and each node is a central vertex of the graph. The topology shown in Figure 5.10 consists of 12 links: each node has degree 3 and requires a minimum of 3 nodes to fail to become disconnected. The robust network precludes use of a channel filter for distributed data fusion, because it violates the tree topology. The network robustness makes it difficult to track common information during fusion without data tagging and generalized data fusion for unknown correlations will be used.

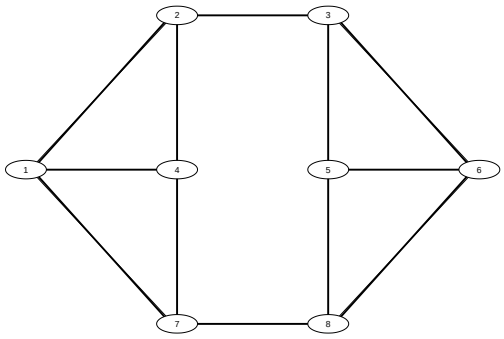

Figure 5.10: Network topology for the generalized distributed mapping experiments. The graph is optimal for 8 nodes and has an equivalent node and link connectivity and requires 3 nodes to fail to become disconnected.

The individual sensor nodes collect and process local scans to build a local occupancy grid map that will be updated with map data passed along the network from other agents. An example of map constructed via local updates only is in Figure 5.11, which shows the maps from Agent 1 (Figure 5.8 Red Path). The map is rendered showing the occupied cells falsely colored according to height and the empty cells are shown in a ghost gray; the unknown cells are not shown. The final pose of the robot is shown using a solid red box. The map shows Agent 1 explores only a portion of the map. To enable a full representation of the map, the distributed data fusion techniques are used with the sensor network in Figure 5.10.

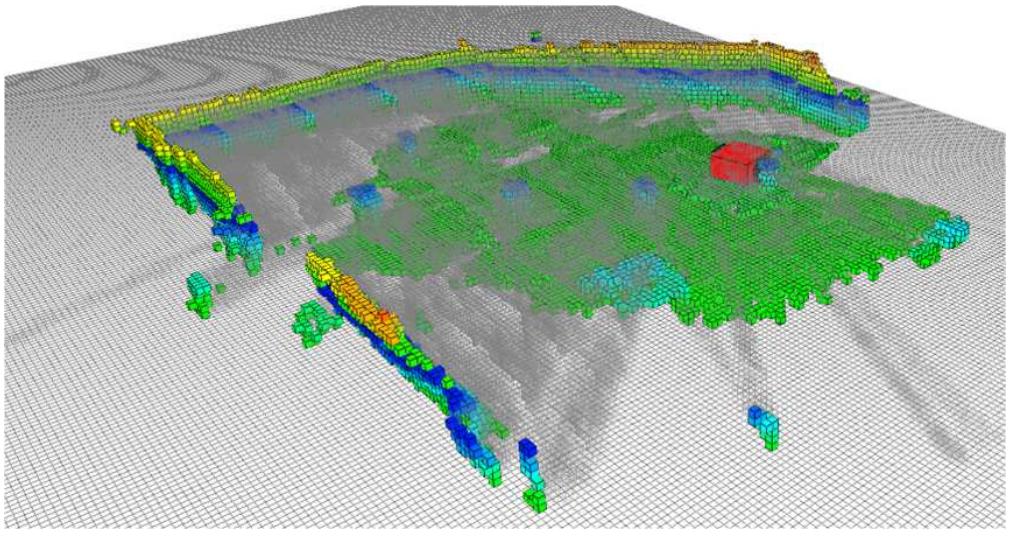

Figure 5.11: 3D Occupancy grid map from Agent 1 using only local updates. The occupied (falsely colored by height) and empty cells (ghost gray) are shown along with the final pose of the robot (red box).

To evaluate the distributed data fusion techniques, the nodes share map information across the bi-directional links shown in Figure 5.10. The agents are required to share the following information for each voxel: the center coordinate and the log-odds probability. This implies each cell requires 32 bytes of data (if all numbers are in double precision) transmitted for each communication. The nodes commu-

nicate aperiodically as they collect information. The final occupancy grid map after Minimum-Information-Loss Fusion is shown in Figure 5.12. The distributed data fusion technique is successfully utilized to build a map that is qualitatively similar to the centralized solution (Figure 5.9). The results for Chernoff Fusion and Entropy Weighted Chernoff Fusion are similar and are not shown.

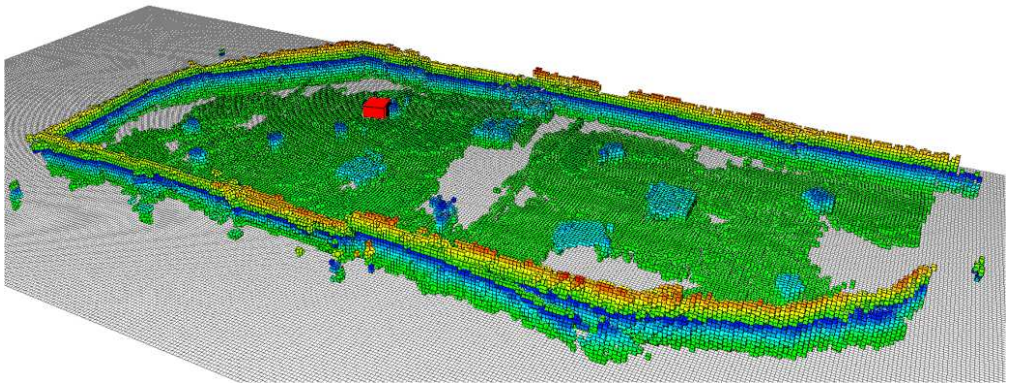

Figure 5.12: 3D Occupancy grid map from Agent 1 after Minimum-Information-Loss Fusion. The occupied (falsely colored by height) are shown along with the final pose of the robot (red box). The map is qualitatively similar to the centralized solution shown in Figure 5.9.

5.4.4 Approximate Information Loss Maps

The approximate information loss $\bar{\mathcal{I}}_{\text{Loss}}$ (5.12) is computed after fusion for each of the different fusion rules. The resulting information loss maps can be used for planning purposes to balance exploration vs verification of cells that may have contained substantial information loss as a result of fusion. The information loss map resulting from Chernoff Fusion is shown in Figure 5.13. The map is falsely colored according to the resulting information loss metric. The minimum values are green and reflect $\bar{\mathcal{I}}_{\text{Loss}} = 0.0$ loss. The maximum values are red and reflect $\bar{\mathcal{I}}_{\text{Loss}} = 0.2$. This reflects the maximum range of values according to the empirical CDF for

the different fusion rules (Figure 5.4). The areas where the largest information loss occurs when fusing with information coming from remote nodes clashes with the content estimate locally at Agent 1 and are in the left hand portion of the map. The areas of information loss to the right hand portion of the map are resulting from the sequential application of the fusion rule as information is received on the three links connected to Agent 1.

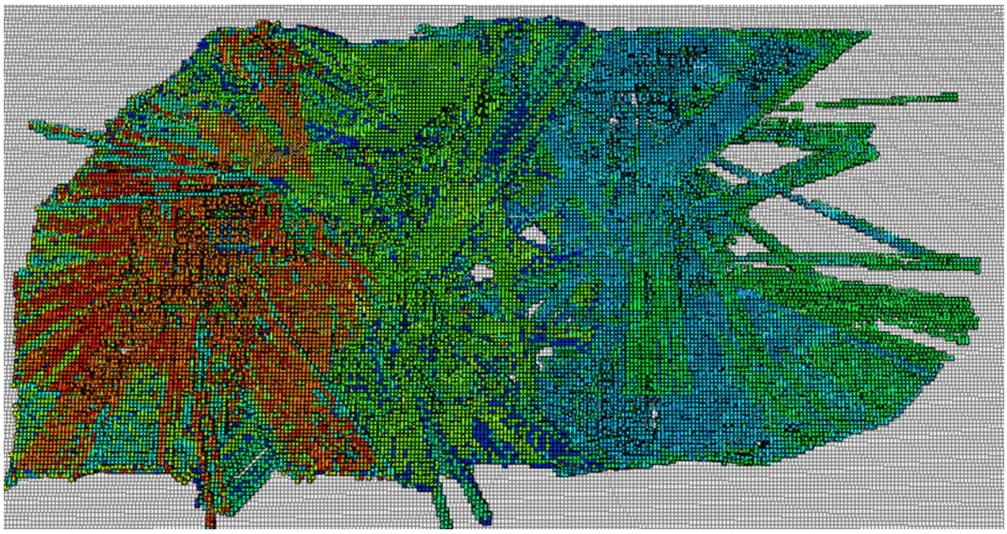

Figure 5.13: Approximate information loss $\bar{\mathcal{I}}_{\text{LOSS}}$ on the occupancy grid after Chernoff Fusion for Agent 1.

The resulting information loss map for Entropy Weighted Chernoff Fusion at Agent 1 is shown in Figure 5.14. The false coloring is the same as the Chernoff Fusion (Figure 5.13). The results shows a decrease in the information loss as a result of Entropy Weighted Chernoff Fusion. The resulting fused map (although not shown) is similar to Figure 5.11. This suggests that the Entropy Weighted Chernoff Fusion will result in a more efficient exploration and verification of the environment, because the resulting map is similar with a reduced information loss. However, the Minimum-Information-Loss Fusion rule is significantly better than either approach in terms of reducing the need for verification due to potential

information loss.

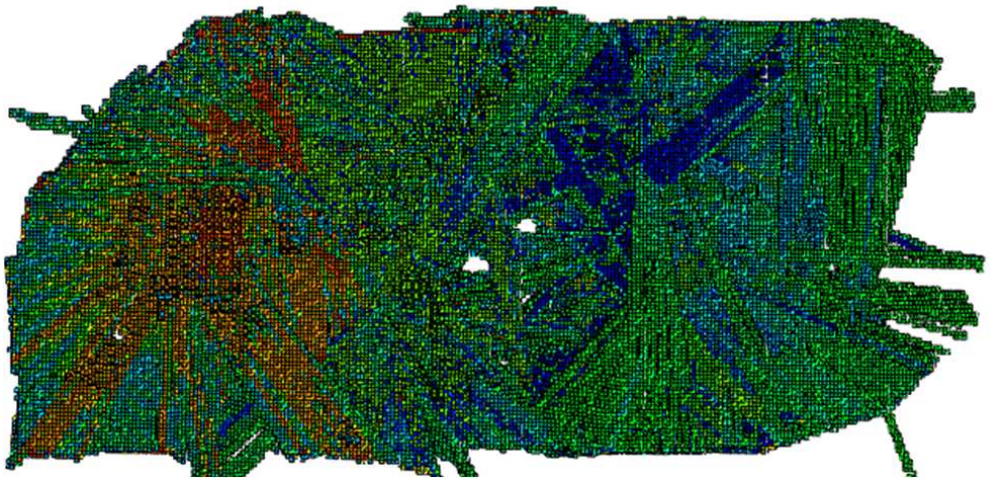

Figure 5.14: Approximate information loss $\bar{\mathcal{I}}_{\text{Loss}}$ on the occupancy grid after Entropy Weighted Chernoff Fusion for Agent 1.

The resulting information loss map for Minimum-Information-Loss Fusion at Agent 1 is shown in Figure 5.15. The false coloring is the same as the Chernoff Fusion (Figure 5.13). The results show a dramatic improvement in the potential information loss, because the fusion rule has zero loss for 50% of the possible combinations of Bernoulli distributions. The Minimum-Information-Loss Fusion rule generates consistent and quality occupancy grid maps and has the lowest information loss. Therefore, this fusion rule is the best for distributed data fusion of occupancy grid maps.

5.5 Conclusions

The generalized weighted exponential product approach for fusing two arbitrary distributions with unknown correlation was addressed. The common approach for finding the fusion weight is to equate the Kullback-Leibler divergence between

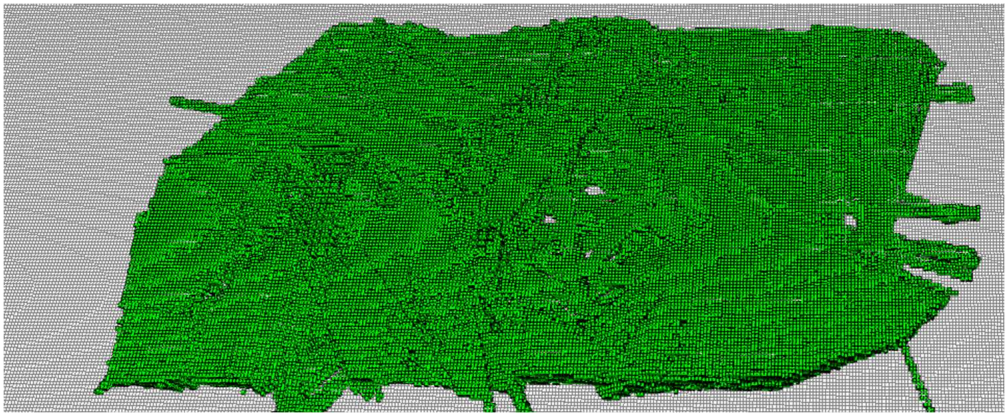

Figure 5.15: Approximate information loss $\bar{\mathcal{I}}_{\text{Loss}}$ on the occupancy grid after Minimax-Loss Fusion for Agent 1.

the fused distribution and the two constituent distribution; this leads to Chernoff Fusion. Chernoff Fusion does not account for unequal sources of information content used to estimate the two constituent distributions in determining the fusion rule. To account for this potential disparity, but without explicitly counting the number of measurements used to estimate the distribution, the Entropy Weighted Chernoff Fusion is developed as a consistent, but conservative fusion approach. Next, a metric for the information loss as a result of WEP fusion is derived and an approximation developed and shown to be the Kullback-Leibler divergence between the Naive Bayes and fused distribution. A second novel fusion rule Minimum-Information-Loss Fusion is developed to minimize the approximate information loss. An in-depth analysis of the three fusion approaches is performed for Bernoulli distributions because they are simple, the fused density, optimal fusion weights and objective functions are computed in closed form. The fusion of multi-robot occupancy grids for agents in an optimally connected sensor network is performed as an application of the generalized data fusion schemes derived.

Acknowledgments

The authors would like to thank Aaron Nathan and the members of the Cornell University Autonomous Systems Laboratory who made data collection and visualization software possible. This work is supported under the Northrop Grumman Electronic Systems Scholars Program.

CHAPTER 6

CONCLUSION

This thesis addressed the issues associated with enabling an always-on and always collaborating network of mobile robots by examining data fusion and distributed robotic perception. The question of what is the best way to fuse and share information collected from network mobile robots was examined through theoretical development, empirical analysis and applied experiment.

A methodology for multi-sensor fusion and multi-modal estimation was developed. A Real-Time implementation of a general Gaussian Sum filter was developed to perform localization in a known map using multi-modal vision measurements in the absence of GPS with map relative measurements. The RT-GSF uses Runnalls' condensation technique to maintain the number of Gaussian components in the posterior density over time; varying the number of terms according to the MMSE state covariance. During experiment, the RT-GSF remains converged with a precise global position estimate over an extended 32 minute GPS blackout. The RT-GSF requires a multi-modal posterior distribution to avoid divergence, although a lower number of mixands in the posterior is required to solve the localization problem. The RT-GSF performs statistically better than the particle filter; in addition, using additional particles does not improve the filter performance due to complex interaction with the measurement hypothesis tests. The RT-GSF performance improvements are driven by time segments where the PF converges to the incorrect road lane. This is typically the result of a misrepresentation of the posterior distribution, which is further shown by the PF continuously rejecting vision measurements after divergence. The PF posterior distribution is inefficient in that it can be sufficiently represented with a small number of terms in a Gaus-

sian mixture, regardless of the number of particles used in the filter. The largest area of complexity in the PF posterior distribution is located around corners in the map, where few map-aided measurements are available. The RT-GSF and PF computation time both scale quadratically, but the RT-GSF scales at a slower rate. The RT-GSF provides an effective multi-sensor and multi-modal estimation scheme that enables networked robots to efficiently collaborate after self-localization.

A technique for sharing and fusing rich probabilistic environment representations was developed for tree and arbitrary connected networks. A grid-based terrain mapping algorithm that maintains a probabilistic accuracy assessment of the height estimate in each grid-cell is successfully developed and demonstrated in laboratory experiment in the distributed data fusion paradigm. Two approaches were developed for distributed data fusion using sufficient statistics of the cell height Gaussian Mixture computed for each grid-cell in the map. First, the exact solution is recovered through tracking of the common information in the sufficient statistic through use of the channel filter when the network is tree connected. Two information metrics were introduced: the Association Probability Metric (APM) and Cumulative Entropy Metric (CEM) to qualitatively evaluate the distributed data fusion schemes. Second, an approach for fusion in arbitrary networks was derived that uses weighted combination of the sufficient statistic set of the Gaussian Mixture for each grid cell. The fusion rule operates on the sufficient statistics of the Gaussian Mixture height distribution and provides a unique resultant Gaussian Mixture mean and covariance. The fusion rule is shown to be quantitatively consistent, but conservative in terms of the APM and CEM using an optimally connected network. Three different techniques for computing the fusion weight are presented for the generalized fusion. The fixed weight and minimum variance weight computations are shown to discount the most information according to the

APM in the network, but still have convergence to the global solution in terms of the CEM. The probabilistic weight computation conserves the most information according to the APM in the network compared to the other techniques, which encourages sensor nodes to explore disjoint areas of the map. The conservative fusion rule is shown to discount information available in the sensor network and a bound of the maximum information loss for each grid cell is derived. The cumulative maximum information loss is shown to be a loose bound, because it is an accumulation of repeated over-estimates. The maximum loss is shown on the terrain map and demonstrates the dichotomy that repeated fusion of the same grid-cells leads to reduced variance, but increases the apparent information loss. The fusion of rich probabilistic environment representations in tree and arbitrary connected networks is a step towards multiple robots coordinating and efficiently exploring and collaborating to share knowledge about their current environment.

An algorithm to exploit sensor modality for higher-level tasks such as object tracking and path planning was developed. An algorithm to segment textured dense 3D point clouds was presented and demonstrated in a complex urban environment. Textured dense point clouds are generated from interpolating sparse laser range finder data constrained by an aligned optical image through a Markov Random Field. The dense point cloud is found using efficient matrix solver routines that enable near real-time performance. The segmentation is performed using an efficient and deterministic algorithm that utilizes the pixel intensity and 3D point geometry data. The algorithm is demonstrated in a complex urban environment which shows the successful ability to segment the road surface as a coherent cluster. The benefit of using the dense range data for interpolation is demonstrated by examining the segmentation performance with the sparse range data alone and augmenting the sparse range data with pixel intensity and surface normal estimates

prior to segmentation. The exploitation of different sensor modalities enable mobile robots to maximize the perception of a given environment and make available rich data for higher-level algorithms. The higher-level algorithms can operate collaboratively and maximize utilization of the information extracted from the low-level sensors.

Consistent data fusion with unknown correlation in robust sensor networks was addressed to mitigate the rumor propagation problem. The generalized weighted exponential product approach for fusing two arbitrary distributions with unknown correlation was examined. The common approach for finding the fusion weight is to equate the Kullback-Leibler divergence between the fused distribution and the two constituent distribution; this leads to Chernoff Fusion. Chernoff Fusion does not account for unequal sources of information content used to estimate the two constituent distributions in determining the fusion rule. To account for this potential disparity, but without explicitly counting the number of measurements used to estimate the distribution, the Entropy Weighted Chernoff Fusion is developed as a consistent, but conservative fusion approach. Next, a metric for the information loss as a result of WEP fusion is derived and an approximation developed and shown to be the Kullback-Leibler divergence between the Naive Bayes and fused distribution. A second novel fusion rule Minimum-Information-Loss Fusion is developed to minimize the approximate information loss. An in-depth analysis of the three fusion approaches is performed for Bernoulli distributions because they are simple, the fused density, optimal fusion weights and objective functions are computed in closed form. The fusion of multi-robot occupancy grids for agents in an optimally connected sensor network is performed as an application of the generalized data fusion schemes derived. The generalized data fusion schemes enable consistent fusion of environment representations within a network of mobile

robots. The sharing and fusion of environmental representations enables collaborative robotics for exploration and planning.

As mobile robots become more ubiquitous, the expectations on their capabilities will increase. The capability of any one robot is expanded via collaboration with other robots. The collaboration possible includes sharing knowledge about the environment which the robots are operating or past experiential information that proves helpful for the current task. The collaboration with heterogeneous robots is going to be enabled with schemes that operate on ad-hoc networks that utilize the existing data network infrastructure, including Wifi and cellular networks. The usefulness of sharing information relies on the ability to integrate diverse information consistently and provides a limitless potential to assist and enhance the lives of humans who operate alongside the robots.

BIBLIOGRAPHY

Alspach, D. and Sorenson, H. (1972). Nonlinear bayesian estimation using gaussian sum approximations. *Automatic Control, IEEE Transactions on*, 17(4):439–448. 0018-9286.

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal filtering*. Prentice-Hall information and system sciences series. Prentice-Hall, Englewood Cliffs, N.J.

Andreasson, H., Triebel, R., and Lilienthal, A. (2007). Non-iterative vision-based interpolation of 3d laser scans. *Computational Intelligence (SCI)*, 76:83–90.

Ani Hsieh, M., Cowley, A., Keller, J. F., Chaimowicz, L., Grocholsky, B., Kumar, V., Taylor, C. J., Endo, Y., Arkin, R. C., Jung, B., Wolf, D. F., Sukhatme, G. S., and MacKenzie, D. C. (2007). Adaptive teams of autonomous aerial and ground robots for situational awareness. *Journal of Field Robotics*, 24(11-12):991–1014.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 50(2):174–188. 1053-587X.

Awerbuch, B., Bar-Noy, A., and Gopal, M. (1994). Approximate distributed bellman-ford algorithms. *Communications, IEEE Transactions on*, 42(8):2515–2517. 0090-6778.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): part ii. *Robotics and Automation Magazine, IEEE*, 13(3):108–117. 1070-9932.

Bailey, T., Julier, S., and Agamennoni, G. (2011). On conservative fusion of information with unknown non-gaussian dependence. Technical report, Austrailian Center for Field Robotics.

Bar-Shalom, Y., Li, X., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, Inc, New York.

Bar-Shalom, Y. and Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(Sept. 1975):451–460.

Bares, J., Hebert, M., Kanade, T., Krotkov, E., Mitchell, T., Simmons, R., and Whittaker, W. (1989). Ambler: an autonomous rover for planetary exploration. *Computer*, 22(6):18–26. 0018-9162.

Besl, P. J. and Jain, R. C. (1988). Segmentation through variable-order surface fitting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(2):167–192. 0162-8828.

Birk, A. and Carpin, S. (2006). Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, 94(7):1384–1397. 0018-9219.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer-Verlag New York, Inc.

Carpin, S. (2008). Fast and accurate map merging for multi-robot systems. *Autonomous Robots*, 25(3):305–316. 10.1007/s10514-008-9097-4.

Carpin, S., Birk, A., and Jucikas, V. (2005). On map merging. *Robotics and Autonomous Systems*, 53(1):1–14. 0921-8890 doi: 10.1016/j.robot.2005.07.001.

Ceruti, M., Wright, T., Powers, B., and McGirr, S. (2006). Data pedigree and strategies for dynamic level-one sensor data fusion. In *Information Fusion, 2006 9th International Conference on*.

Chang, K., Chee-Yee, C., and Mori, S. (2010). Analytical and computational evaluation of scalable distributed fusion algorithms. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(4):2022–2034. 0018-9251.

Chang, K. C., Chee-Yee, C., and Mori, S. (2008). On scalable distributed sensor fusion. In *Information Fusion, 2008 11th International Conference on*, pages 1–8.

Cheng, Y. and Singh, T. (2007). Efficient particle filtering for road-constrained target tracking. *Aerospace and Electronic Systems, IEEE Transactions on*, 43(4):1454–1469. 0018-9251.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*, volume 6. Wiley Online Library.

Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746. 1053-587X.

Cui, Y. and Ge, S. S. (2003). Autonomous vehicle positioning with gps in urban canyon environments. *Robotics and Automation, IEEE Transactions on*, 19(1):15–25. 1042-296X.

Dekker, A. H. and Colbert, B. D. (2004). Network robustness and graph topology. 979965 359-368.

Diebel, J. and Thrun, S. (2006). An application of markov random fields to range sensing. *Advances in Neural Information Processing Systems*, 18:291.

Douc, R. and Cappe, O. (2005). Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69.

Doucet, A. and Johansen, A. M. (2009). *A tutorial on particle filtering and smoothing: Fifteen years later.* Oxford Handbook of Nonlinear Filtering. Oxford University Press.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57. 0018-9162.

Farrell, W. J. and Ganesh, C. (2009). Generalized chernoff fusion approximation for practical distributed data fusion. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 555–562.

Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181. 10.1023/B:VISI.0000022288.19776.77.

Fletcher, L., Teller, S., Olson, E., Moore, D., Kuwata, Y., How, J., Leonard, J., Miller, I., Campbell, M., Huttenlocher, D., Nathan, A., and Kline, F.-R. (2008). The mit-cornell collision and why it happened. *Journal of Field Robotics Special Issue on the DARPA Urban Challenge*, 25(10):775–807.

Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184. 0364-9059.

Fosbury, A. M., Singh, T., Crassidis, J. L., and Springen, C. (2007). Ground target tracking using terrain information. In *Information Fusion, 2007 10th International Conference on*, pages 1–8.

Fouque, C. and Bonnifait, P. (2008). Tightly-coupled gis data in gnss fix computations with integrity testing. *International Journal of Intelligent Information and Database Systems*, 2(2):167–186.

Fournier, J., Ricard, B., and Laurendeau, D. (2007). Mapping and exploration of complex environments using persistent 3d model. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 403–410.

Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., and Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339. 0018-9219.

Fregene, K., Kennedy, D., Madhavan, R., Parker, L. E., and Wang, D. (2005). A class of intelligent agents for coordinated control of outdoor terrain mapping ugvs. *Engineering Applications of Artificial Intelligence*, 18(5):513–531. 0952-1976 doi: DOI: 10.1016/j.engappai.2004.12.007.

Gachter, S., Nguyen, V., and Siegwart, R. (2006). Results on range image segmentation for service robots. In *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*, pages 53–53.

Genest, C. and Zidek, J. V. (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135. 08834237 ArticleType: research-article / Full publication date: Feb., 1986 / Copyright 1986 Institute of Mathematical Statistics.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113. 0956-375X.

Gould, S., Baumstarck, P., Quigley, M., Ng, A. Y., and Koller, D. (2008). Integrating visual and range data for robotic object detection. In *European Conference on Computer Vision*, volume M2SFA2, Marseille, France.

Grime, S. and Durrant-Whyte, H. F. (1994). Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849–863. 0967-0661 doi: 10.1016/0967-0661(94)90349-2.

Grime, S., Durrant-Whyte, H. F., and Ho, P. (1992). Communication in decentralized data-fusion systems. In *American Control Conference, 1992*, pages 3299–3303.

Harrison, A. and Newman, P. (2009). Image and sparse laser fusion for dense scene reconstruction. In *7th International Conference on Field and Service Robotics*, Cambridge, Massachusetts.

Huber, M. F., Bailey, T., Durrant-Whyte, H., and Hanebeck, U. D. (2008). On entropy approximation for gaussian mixture random vectors. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 181–188.

Hurley, M. B. (2001). An information-theoretic justification for covariance intersection and its generalization. Technical report, MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB.

Hurley, M. B. (2002). An information theoretic justification for covariance intersection and its generalization. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 1, pages 505–511 vol.1.

Intel®(2011). Math kernel library 10.2 (http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation/).

Julier, S. (2006). An empirical study into the use of chernoff information for robust, distributed fusion of gaussian mixture models. In *Information Fusion, 2006. FUSION '06. 9th International Conference on*.

Julier, S. J. (2009). Estimating and exploiting the degree of independent information in distributed data fusion. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 772–779.

Julier, S. J. and Uhlmann, J. K. (1997). A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference, 1997. Proceedings of the 1997*, volume 4, pages 2369–2373 vol.4.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3206–3211.

Klasing, K., Wollherr, D., and Buss, M. (2008). A clustering method for efficient segmentation of 3d laser data. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 4043–4048.

Kleiner, A. and Dornhege, C. (2007). Real-time localization and elevation mapping within urban search and rescue scenarios: Field reports. *Journal of Field Robotics*, 24(8-9):723–745.

Konolige, K., Fox, D., Limketkai, B., Ko, J., and Stewart, B. (2003). Map merging for distributed robot navigation. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 212–217 vol.1.

Kotecha, J. H. and Djuric, P. M. (2003a). Gaussian particle filtering. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 51(10):2592–2601. 1053-587X.

Kotecha, J. H. and Djuric, P. M. (2003b). Gaussian sum particle filtering. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 51(10):2602–2612. 1053-587X.

Kozak, M. C. and Maybeck, P. S. (2006). Enhanced multiple model tracker based on gaussian mixture reduction for a maneuvering target in clutter. In *Signal*

and Data Processing of Small Targets 2006, volume 6236, pages 62360I–11, Orlando (Kissimmee), FL, USA. SPIE.

Kraetzschmar, G. K., Gassull, G. P., and Uhl, K. (2004). Probabilistic quadtrees for variable-resolution mapping of large environments. In Ribeiro, M. and Victor, S., editors, *Proceedings of the 5th IFAC/EURON symposium on intelligent autonomous vehicles*, Lisbon, Portugal.

Kronhamn, T. R. (1998). Bearings-only target motion analysis based on a multihypothesis kalman filter and adaptive ownship motion control. *Radar, Sonar and Navigation, IEE Proceedings* -, 145(4):247–252. 1350-2395.

Kwok, N. M., Dissanayake, G., and Ha, Q. P. (2005). Bearing-only slam using a sprt based gaussian sum filter. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1109–1114.

Lemaire, T. and Lacroix, S. (2007). Slam with panoramic vision. *Journal of Field Robotics*, 24(1-2):91–111. 1556-4967.

Levinson, J., Montemerlo, M., and Thrun, S. (2007). Map-based precision vehicle localization in urban environments. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA.

Liggins, M. E., I., Chee-Yee, C., Kadar, I., Alford, M. G., Vannicola, V., and Thomopoulos, S. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107. 0018-9219.

Lim, E. H. and Suter, D. (2008). Multi-scale conditional random fields for oversegmented irregular 3d point clouds classification. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–7.

Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping.

Mahler, R. (2000). Optimal/robust distributed data fusion: a unified approach. *Proceedings of SPIE*, 4052(1):128–132.

Makarenko, A., Brooks, A., and Kaupp, T. (2007). On the benefits of making robotic software frameworks thin. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'07)*, San Diego CA, USA.

Makarenko, A. and Durrant-Whyte, H. (2006). Decentralized bayesian algorithms for active sensor networks. *Information Fusion*, 7(4):418–433. 1566-2535 doi: DOI: 10.1016/j.inffus.2005.09.010.

Martin, D., Fowlkes, C., Doron, T., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–416.

Martinez-Cantin, R., Castellanos, J. A., and de Freitas, N. (2007). Multi-robot marginal-slam. In *International Joint Conference on Artificial Intelligence*, Hyderabad, India.

Maybeck, P. S. and Smith, B. D. (2005). Multiple model tracker based on gaussian mixture reduction for maneuvering targets in clutter. In *Information Fusion, 2005 8th International Conference on*, volume 1, page 8 pp.

Miller, I. and Campbell, M. (2008). Particle filtering for map-aided localization in sparse gps environments. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1834–1841.

Miller, I., Campbell, M., and Huttenlocher, D. (2011). Map-aided localization in

sparse gps environments using vision and particle filtering. *Journal of Field Robotics*. Accepted and to appear.

Miller, I., Campbell, M., and Huttenlocher, D. (In Submission). Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes. *Robotics, IEEE Transactions on*.

Miller, I., Campbell, M., Huttenlocher, D., Kline, F.-R., Nathan, A., Lupashin, S., Catlin, J., Schimpf, B., Moran, P., Zych, N., Garcia, E., Kurdziel, M., and Fujishima, H. (2008a). Team cornell's skynet: Robust perception and planning in an urban environment. *Journal of Field Robotics*, 25(8):493–527.

Miller, I. and Campbell, M. E. (2006). A mixture-model based algorithm for real-time terrain estimation. *J. Robot. Syst.*, 23(9):755–775.

Miller, I., Schimpf, B., Campbell, M., and Leyssens, J. (2008b). Tightly-coupled gps / ins system design for autonomous urban navigation. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1297–1310.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., and Thrun, S. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597. 1556-4967.

Moravec, H. P. (1996). Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical report, The Robotics Institute Carnegie Mellon University.

Napier, A., Sibley, G., and Newman, P. (2010). Real-time bounded-error pose estimation for road vehicles using vision. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1141–1146.

Nettleton, E. W., Durrant-Whyte, H. F., Gibbens, P. W., and Goektogan, A. H. (2000). Multiple-platform localization and map building. In *Sensor Fusion and Decentralized Control in Robotic Systems III*, volume 4196, pages 337–347, Boston, MA, USA. SPIE.

O'Callaghan, S. T., Ramos, F. T., and Durrant-Whyte, H. (2010). Contextual occupancy maps incorporating sensor and location uncertainty. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3478–3485.

Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system. In *Robotics and Automation, 2011. ICRA '11. IEEE International Conference on*.

Peach, N. (1995). Bearings-only tracking using a set of range-parameterised extended kalman filters. *Control Theory and Applications, IEE Proceedings* -, 142(1):73–80. 1350-2379.

Pfaff, P., Triebel, R., and Burgard, W. (2007). An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *The International Journal of Robotics Research*, 26(2):217–230.

Polak, M., Zhang, H., and Pi, M. (2009). An evaluation metric for image segmentation of multiple objects. *Image and Vision Computing*, 27(8):1223–1227. 0262-8856 doi: DOI: 10.1016/j.imavis.2008.09.008.

Rendas, M. J. and Leita, J. M. (2010). Rumor-robust distributed data fusion. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, pages 230–235.

Rivadeneyra, C. and Campbell, M. (2011). Probabilistic multi-level maps from lidar data. *The International Journal of Robotics Research*.

Roth, G. and Wibowo, E. (1997). An efficient volumetric method for building close triangular meshes from 3-d image and point data. In *Graphics Interface 97*, pages 173–180, Kelowna, B.C., Canada.

Runnalls, A. R. (2007). Kullback-leibler approach to gaussian mixture reduction. *Aerospace and Electronic Systems, IEEE Transactions on*, 43(3):989–999. 0018-9251.

Rusu, R. B., Blodow, N., Marton, Z., Soos, A., and Beetz, M. (2007). Towards 3d object maps for autonomous household robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3191–3198.

Ryde, J. and Hu, H. (2006). Mutual localization and 3d mapping by cooperative mobile robots. In *9th International Conference on Intelligent Autonomous Systems*, pages 217–224, The University of Tokyo, Tokyo, Japan. IOS Press.

Salmond, D. J. (1988). Mixture reduction algorithms for uncertain tracking. Technical Report 88004, ROYAL AEROSPACE ESTABLISHMENT.

Schoenberg, J. and Campbell, M. (2012a). Consistent generalized data fusion for multiple robot occupancy grid mapping. *International Journal of Robotics Research*, In Preparation.

Schoenberg, J. and Campbell, M. (2012b). Generalized data fusion for distributed terrain estimation. *International Journal of Robotics Research*, In Review.

Schoenberg, J., Miller, I., and Campbell, M. (2012). Posterior representation with a multi-modal likelihood using the gaussian sum filter for localization in a known map. *Journal of Field Robotics*, To Appear.

Schoenberg, J. R. and Campbell, M. (2009). Distributed terrain estimation using

a mixture-model based algorithm. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 960–967.

Schoenberg, J. R., Campbell, M., and Miller, I. (2009). Localization with multi-modal vision measurements in limited gps environments using gaussian sum filters. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1423–1428.

Schoenberg, J. R., Nathan, A., and Campbell, M. (2010). Segmentation of dense range information in complex urban scenes. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2033–2038.

Se, S., Barfoot, T., and Jasiobedzki, P. (2005). Visual motion estimation and terrain modeling for planetary rovers. In Battrick, B., editor, *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, pages 101–109, Munich, Germany. ESA SP-603.

Sola, J., Monin, A., Devy, M., and Lemaire, T. (2005). Undelayed initialization in bearing only slam. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2499–2504.

Sorenson, H. W. and Alspach, D. L. (1971). Recursive bayesian estimation using gaussian sums. *Automatica*, 7(4):465–479.

Syed, E. and Cannon, M. (2005). Map-aided gps navigation. *GPS World*, 16(11):39–44.

Thompson, P. R. and Durrant-Whyte, H. (2010). Decentralised data fusion in 2-tree sensor networks. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8.

Thrun, S. (2001). A probabilistic on-line mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.

Thrun, S. (2002). Robotic mapping: A survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in a New Millenium*. Morgan Kaufmann.

Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 1, pages 321–328 vol.1.

Thrun, S., Martin, C., Yufeng, L., Hahnel, D., Emery-Montemerlo, R., Chakrabarti, D., and Burgard, W. (2004). A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *Robotics and Automation, IEEE Transactions on*, 20(3):433–443. 1042-296X.

Triebel, R., Pfaff, P., and Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 22762282, Beijing, China.

Upcroft, B., Kumar, S., Ridley, M., Ong, L., and Durrant-Whyte, H. (2004). Fast re-parameterisation of gaussian mixture models for robotics applications. In *Proceedings of the 2004 Australasian Conference on Robotics and Automation*, Canberra, Australia. ANSER-II.

Wijesoma, W. S., Lee, K. W., and Ibanez-Guzman, J. (2005). Motion constrained simultaneous localization and mapping in neighbourhood environments. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1085–1090.

Williams, J. L. (2003). *Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter*. PhD thesis, Air Force Institute of Technology.

Williams, J. L. and Maybeck, P. S. (2003). Cost-function-based gaussian mixture reduction for target tracking. In *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, volume 2, pages 1047–1054.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. (2010). Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*.